A Multigroup Eigenfunction Expansion Technique
For Selecting Poison Distributions
For Optimum Reactor Power Distribution

By

STEVEN MICHAEL LONG

A DISSERTATION PRESENTED TO THE GRADUATE
COUNCIL OF THE UNIVERSITY OF FLORIDA IN PARTIAL
FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA
1972

This dissertation is dedicated to my lovely wife,

Diane,

who ungrudgingly shared the effort of completing

the beast during our first months of marriage.

ACKNOWLEDGEMENTS

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

A	energy per fission (in appropriate units)
a_i	expansion coefficient associated with function i
$\bar{a}$	vector representing a set of expansion coefficients
[A]	matrix on left-hand side of various eigen-value formulations of the criticality equation
B_z^2	axial geometric buckling
[B]	matrix on the right-hand side of the general eigenvalue formulation of the criticality equation
$C(k,i,j)$	coefficient of the flux in group k at point j in the gradient term of the equation for the flux in group k at point i
$D^k(\bar{r})$	diffusion coefficient function in energy group k
$D(k,j)$	diffusion coefficient at point j in group k
E	weighted scalar squared error optimization parameter
$\mathcal{E}$	vector of error components
$F(m,i,[N])$	the scalar product of the poison eigenvector i and the adjoint vector m, weighted by the operator $[N][\sigma_p]$
[F]	matrix with elements $F(m,i,[N])$
G	number of control points
H	number of fueled points
I	number of eigenvectors
i	preferred index for designating a particular eigenvector

$[I]$	the identity matrix
J	number of spatial points
$J_o(r)$	zeroth order Bessel function of the first kind
$[J]$	the Jacobian matrix
j	preferred index for designating a spatial point
K	number of energy groups
K_{eff}	reactivity of a reactor
k	preferred index for designating an energy group
$[L]$	matrix containing the neutron loss terms of the criticality equation
M	number of vectors used in expansion
m	preferred index for designating a particular adjoint vector
$N(\overline{r})$	poison distribution function
$N(j)$	poison concentration at point j
$\overline{N}$	vector with elements N(j)
$[N]$	diagonal matrix with elements N(j)
n	auxiliary index for indicating energy group
$[0]$	zero matrix
$P(\overline{r})$	power distribution function
$P*(\overline{r})$	ideal power function
$P(j)$	power density at point j
P_i	power vector associated with eigenvector ϕ_i
$P*$	ideal power vector
$[Q]$	thermal poison matrix
r	radius
$\overline{r}$	general position vector

$s^k(\overline{r})$	spectrum function relating the flux in group k to the flux in group K
s_i	fast/thermal flux ratio for eigenfunction i (spatially constant)
$[S]$	matrix containing the fission source terms of the criticality equation
$\overline{V}$	arbitrary vector
$V(j)$	fraction of reactor volume associated with mesh point j
$[V]$	diagonal matrix containing the inverse velocity terms of the Laplace transformed kinetic equation
$[W]$	diagonal matrix with elements $V(j)$
X	distance from centerplane in slab geometry
α_i	eigenvalue associated with the eigenfunction i to a general eigenvalue problem
β_i	eigenvalue associated with poison eigenfunction i
$[\beta]$	diagonal matrix with elements β_i
γ_i	eigenvalue associated with the "natural" eigenfunction i
δ_i	Wielandt transformed eigenvalue associated with the mode i
$\delta(i,j)$	Krönecker delta
ϵ	scalar coefficient for the adjustment of the magnitude of a poison distribution
ϵ_i	poison distribution magnitude eigenvalue associated with mode i
n_i	zero number i of the function $J_o(r)$
λ_i	eigenvalue associated with the fission eigenfunction i
μ_i	eigenvalue associated with the thermal poison eigenfunction i
ν	number of neutrons per fission

σ_p^k	poison cross section for a unit concentration
$\sigma_p^k(j)$	equals σ_p^k if point j is a control point, zero otherwise
$[\sigma_p]$	diagonal matrix with elements $\sigma_p^k(j)$
$\Sigma_a^k(\bar{r})$	absorption cross section function for group k
$\Sigma_a(k,j)$	absorption cross section for group k at point j
$\Sigma_f^k(\bar{r})$	fission cross section function for group k
$\Sigma_f(k,j)$	fission cross section for group k at point j
$\Sigma_p^k(\bar{r})$	controllable poison cross section function for group k
$\Sigma_R^k(\bar{r})$	removal cross section function for group k
$\Sigma_R(k,j)$	removal cross section for group k at point j
$\Sigma_s(n{\to}k,j)$	cross section at point j for scattering from group n to group k
$\phi^k(\bar{r})$	flux function for group k
$\phi^*(\bar{r})$	ideal flux function
$\phi(k,j)$	flux at point j in group k
ϕ	flux vector with elements $\phi(k,j)$
ϕ_i	flux eigenvector i
χ^k or $X(k)$	fission spectrum fraction in group k
$\psi(k,j)$	adjoint flux at point j in group k
ψ_m	adjoint flux vector m
ω_i	eigenvalue associated with kinetic mode i

Abstract of Dissertation Presented to the Graduate Council
of the University of Florida in Partial Fulfillment of
the Requirements for the Degree of Doctor of Philosophy

A MULTIGROUP EIGENFUNCTION EXPANSION TECHNIQUE
FOR SELECTING POISON DISTRIBUTIONS
FOR OPTIMUM REACTOR POWER DISTRIBUTION

By

Steven Michael Long

December, 1972

Chairman: Dr. G. Ronald Dalton
Major Department: Nuclear Engineering Sciences

The purpose of this study is to explore the capabilities
of an eigenfunction expansion technique for aiding in the
selection of a poison distribution that will produce the
optimum power distribution in a reactor.   A method is
introduced which is useful in the repeated evaluation of
power distributions resulting from various poison distri-
butions as they are selected for investigation by a numerical
optimization routine. This method is based in multigroup
diffusion theory, and employs the finite difference approxi-
mations to the spatial dimensions.

First, the eigenfunction expansion problem is reviewed
in one-group theory. Then, the multigroup problem is out-
lined and the arbitrariness of the flux-power relationship
in this formulism is discussed. The actual unknowns for the
optimization problem (using a preselected poison material)
are pointed out to be the poison concentrations and the flux
spectral relationships. The inherent nonlinearity of the

problem is discussed for the case where these unknowns are treated directly as variables.

The orthogonality and completeness properties of various sets of eigenvectors are discussed regarding the poison optimization problem. These sets include the natural modes, the fission modes, the time-space separable kinetic modes, the thermal poison modes and the multigroup poison modes.

The multigroup poison eigenvectors are chosen for use in the expansion technique. A Wielandt transformation method for generating this set of eigenvectors is explained, and a computer program, demonstrating the technique, is listed as an appendix. The poison eigenvectors for several two-group, multiregion reactors are presented.

The necessity for solution of the optimum multigroup poison distribution problem by numerical search is discussed in general. The poison eigenvalue formulation of the criticality equation is introduced, and its advantages and weaknesses, as compared to the usual reactivity eigenvalue formulation, are discussed. Two methods are suggested for overcoming the weaknesses. One, a Wielandt transformation method, is conceptually straightforward, but computationally clumsy. The second, a transformation of the set of eigenvalues of the poison eigenvectors, is potentially a better computational device, but is conceptually much more complex. Only the first method is programmed and demonstrated by numerical example.

Sample two-group, one-dimensional optimization searches, using the Gaussian least squared error method, are presented for several examples in order to demonstrate the techniques and phenomena that have been discussed. The results are checked for accuracy with a standard multigroup diffusion theory code. These calculations are then extended to explore the possibility of using incomplete sets of poison eigenvectors for expansion. The results of partial expansions in both underspecified and completely specified optimizations are discussed. Considerable accuracy of partial expansions is shown in the limited class of examples considered (thermal reactors).

Finally, the author presents his thoughts on the potential utility of the method and makes suggestions for future work directed at ultimately incorporating the techniques in a standardized computer code for use as an engineering tool.

CHAPTER I

INTRODUCTION

The purpose of this study is to explore the capabilities of an eigenfunction expansion technique in the problem of reactor power shaping by neutron poison distribution. The mathematical model used for the reactor core is that of multi-group diffusion theory. The spatial dimensions are represented by the finite difference approximation, and the functions used for expansion are the eigenvectors to the problem as expressed in this framework. The author believes that this technique will provide a flexible and realistic representation of the physical problem and lend computational power as well.

The problem of determining the control poison distribution for the optimization of some reactor performance parameter has been treated by many authors, using various methods and mathematical formulisms. Harker[1] used a pseudo two-group diffusion theory that made use of a "fast fission factor", which allowed him to determine a flux distribution from a desired power distribution. He could then calculate the distribution of a thermal poison necessary to achieve criticality for this desired flux shape. Unfortunately, he provided no _a priori_ method of guessing a power distribution

1

which would result in a physically realizable flux shape. Haling[2] and Crowther[3] used a more sophisticated two-group theory and a "backwards iteration technique" to find the flux shape, from which they calculated the necessary distributions of a thermal poison. Crowther applied this "Power Control Method" to the problem of minimizing the peak to average power ratio over core life. Motoda and Kawai[4] applied the Pontryagin Principle of optimal control to the problem of maximizing core lifetime while limiting the power peaking ratio. They used one-group, two-mode formulism to represent a two-region, bare core with control rods ganged in each region. Terney and French[5] used dynamic programming to minimize power peaking over core life, based on a direct flux synthesis method (using two fluxes), for approximating a two-group, two-dimensional reactor with two gangs of control rods. Wade and Terney[6] applied optimal control theory to a generalized set of design objectives, using a one-group spatially nodalized reactor model. Jackson[7] used an eigenfunction expansion technique, based on one-group diffusion theory, to determine the optimum physically realizable power and flux distributions and the poison distribution necessary to maintain these shapes in the critical state. Because of the direct relationship to the subject of this paper, Jackson's work will be described in some detail in Chapter II, which reviews the one-group eigenfunction expansion problem.

Eigenvalue problems are well known to the nuclear engineer in such forms as the reactor criticality equations and the kinetic equations. However, except in Jackson's work and in some kinetics work, little use has been made of the higher modes to these problems. Even for perturbation calculations, only the fundamental eigenfunction (and its adjoint) are in general use for the solution of static problems. Such first moment calculations are only the barest form of eigenfunction expansion.

Investigative interest in eigenfunction expansion techniques is not new, even if not widely spread. In 1960, Foderaro and Garabedian[8] introduced a method for the solution of group diffusion equations which involved expansion of the problem in the eigenfunctions of the Helmholtz equation for the appropriate geometry. An article published in 1962 by Garabedian and Thomas[9] extended the approach to two dimensions. While these methods have the advantages of using well-known sets of tabulated functions and analytically handling the leakage terms within homogeneous regions, they possess a distinct disadvantage. Because these functions are not actually eigenfunctions of the multiregion problem being considered, they require individual (but coupled) expansions for each materials region and require a large number of terms to adequately represent the solution. In 1968, Hara and Shibata[10] combined this approach with variational calculus in a study to optimize total power and

conversion ratio. Their independent variables were region-wise fuel concentrations and region boundary locations. Still these works are best described as expansions in sets of tabulated, orthonormal functions, rather than as modal expansions.

True modal expansion techniques for kinetic problems were explored by Kaplan[11] in 1961. He illustrated techniques using three different sets of eigenfunctions, each generated from a different eigenvalue equation associated with his problem. He introduced the property of "finality", that the value of the expansion coefficients are independent of the order of the expansion, and showed how this property could be achieved for certain kinetics problems. He termed the set of eigenfunctions having this property for a particular problem the "natural modes" for that problem. In 1966, Kaplan and Yasinsky[12] used an analysis of the eigenvalues of the natural modes to predict the behavior of xenon oscillations. Henry,[13] in 1963, extended modal approximations to the nonseparable space-time kinetics problems, defining what he termed the "inhour modes". These occur in clusters of seven, with similar spatial shapes within clusters.

In 1963, the MULE computer code[14] became available for computing three particular types of eigenvector sets to two-group diffusion theory reactor models. These eigenfunctions sets included: 1) the "λ modes" or "fission eigenfunctions"

where λ is the inverse reactivity for the mode, 2) the "μ modes", where μ is the concentration multiplier for a given normalized distribution of thermal poison, and 3) the "ω modes" or eigenfunctions to the Laplace transformed, time-space separable kinetics problem. As will be discussed in Chapter IV, with the exception of the ω modes, these modes are degenerate in the energy dimension, and are limited, for expansion purposes, to expansions of the spatial dimensions, alone.

In this study, an expansion technique using a set of eigenvectors capable of representing both the energy and space dimensions is developed. A method for generating such eigenvectors will be presented, and its use demonstrated. These vectors and the expansion technique will then be displayed in the determination of the poison distributions and concentrations necessary for criticality in the optimum realizable flux-power shapes for several example reactors.

CHAPTER II

THE ONE-GROUP EIGENFUNCTION EXPANSION PROBLEM

Although a treatment of the one-group problem would seem to be the simplest case of a general multigroup treatment, such generalization is not necessarily possible. Because the one-group formulism does not contain the energy variable, it is amenable to a quite different treatment than is a multigroup problem. As will be discussed in Chapter III, this is due to the nonlinear nature in which the energy variable enters the poison distribution-power distribution problem. Still, it is instructive to review the one-group treatment for purposes of later comparison and for the purpose of introducing some concepts in a less complicated context. The one-group problem and solution was described by Jackson in reference 7, and his approach will generally be followed in this chapter.

In the one-group formulism, the criticality equation is

$$\nabla \cdot D(\overline{r}) \nabla \phi(\overline{r}) + \nu \Sigma_f(\overline{r}) \phi(\overline{r}) - \Sigma_a(\overline{r}) \phi(\overline{r}) - \Sigma_p(\overline{r}) \phi(\overline{r}) = 0 \qquad (1)$$

plus the boundary conditions on $\phi(\overline{r})$ and/or its first derivative. Standard notation is used except that $\Sigma_p(\overline{r})$ is the macroscopic cross section of the controllable poison.

The power-flux relationship is

$$P(\bar{r}) = A\Sigma_f(\bar{r})\phi(\bar{r}), \tag{2}$$

where A is the energy per fission in appropriate units. At
first glance, it may appear that, once a desired power
shape, $P^*(\bar{r})$, is chosen, the resultant flux could be found
directly from equation (2) as

$$\phi^*(\bar{r}) = \frac{P^*(\bar{r})}{A\Sigma_f(\bar{r})}, \tag{3}$$

and that the desired distribution of control poison follows
directly from equation (1) as

$$\Sigma_p(\bar{r}) = \frac{\nabla \cdot D(\bar{r})\nabla\phi^*(\bar{r})}{\phi^*(\bar{r})} + \nu\Sigma_f(\bar{r}) - \Sigma_a(\bar{r}). \tag{4}$$

This is the approach taken by Harken,[1] Haling[2] and
Crowther.[3] As Jackson[7] pointed out, however, most
reactors of interest will have step variations in $\Sigma_f(\bar{r})$,
due to the practice of regionwise fuel loading. Thus, the
ideal flux as defined by equation (3) must also have step
variations. Physically, the flux cannot behave in this man-
ner. As represented by diffusion theory, the neutron flux,
$\phi(\bar{r})$, and the neutron current, $D(\bar{r})\nabla\phi(\bar{r})$, are constrained
to be continuous.

Jackson proposed an eigenfunction expansion technique
to get around this problem. Following Kaplan,[11] he chose
the set of "natural modes" to his problem as defined by the
equation,

$$\nabla \cdot D(\overline{r}) \nabla \phi_i(\overline{r}) + \nu \Sigma_f(\overline{r}) \phi_i(\overline{r}) - \Sigma_a(\overline{r}) \phi_i(\overline{r}) = \gamma_i \phi_i(\overline{r}). \qquad (5)$$

These eigenfunctions, the ϕ_i, form a complete, self-adjoint set[7] over the entire reactor. They satisfy all of the boundary and interface conditions required of the real flux, and, thus, any linear sum of these eigenfunctions will also be a solution to equation (1).

A physically realizable approximation to the "ideal flux shape" given by equation (3) can then be constructed from these eigenfunctions. Using the orthogonality relation

$$\int_O^R \phi_i \phi_j d\overline{r} = \delta(i,j), \qquad (6)$$

the expansion coefficients are found by

$$a_i = \int_O^R \phi_i \phi^* d\overline{r}. \qquad (7)$$

These coefficients are found independently of one another, or the order of expansion, I, and thus the expansion has the property of finality. The constructed optimum real flux,

$$\phi_R^*(\overline{r}) = \sum_{i=1}^{I} a_i \phi_i(\overline{r}), \qquad (8)$$

approximates the "ideal flux", ϕ^*, in the least squared errors sense[15] to the degree of expansion, I. Thus, the optimum poison concentrations may be determined from an equation analogous to equation (4), but written for ϕ_R^*. After invoking the eigenvalue relationships of equation (5), this becomes

$$\Sigma_p(\overline{r}) = \frac{\sum\limits_{i=1}^{I} a_i \gamma_i \phi_i(\overline{r})}{\sum\limits_{i=1}^{I} a_i \phi_i(\overline{r})} \quad . \tag{9}$$

As pointed out by Jackson, this approach still has a failing. Because the optimum power shape, $P^*(r)$, is defined only for the reactor core, where $\Sigma_f(\overline{r}) > 0$, but these eigenfunctions are orthogonal only over the complete reactor, the expansion coefficients must contain an ambiguity if a reflector is present. This is due to the arbitrariness of $\phi^*(\overline{r})$ in regions where $\Sigma_f(r) = 0$ and the necessity of integration over those regions. Jackson suggested that the flux in the reflector might either be assumed unchanged from the unpoisoned case, or that it be given a constant value, "treating its magnitude as an adjustable parameter, the adjustment of which might allow the eigenfunction approximation to the ideal in-core flux to be close in that core region in which the nearness of the approximation is considered to be most important."[7:24]

Neither suggestion seems particulary realistic. The second has the added disadvantage of somewhat linking the solutions of the expansion coefficients by evaluating them together during the optimization search for the reflector flux magnitude. This not only violates the conditions necessary for the property of finality, it also introduces an added complexity which seems to destroy much of the power of the method.

Recognizing a better alternative himself, Jackson recommended the exploration of other sets of eigenfunctions which are complete and orthogonal over just the fueled portions of the reactor. He suggested a set he called the "fission eigenfunctions", given by the equation

$$\nabla \cdot D(\overline{r}) \nabla \phi_i(\overline{r}) - \Sigma_a(\overline{r}) \phi_i(\overline{r}) = -\lambda_i \nu \Sigma_f(\overline{r}) \phi_i(\overline{r}), \tag{10}$$

and showed that they satisfy the orthogonality relation

$$\int_0^R \phi_i(\overline{r}) \nu \Sigma_f(\overline{r}) \phi_j(\overline{r}) dr = \delta(i,j). \tag{11}$$

Because of the weighting factor, $\Sigma_f(\overline{r})$, fluxes in unfueled regions have no effect on the computation of the expansion coefficients for this vector set, as they carry zero weight.

The expansion coefficients are thus found from

$$a_i = \int_0^R \phi_i(\overline{r}) \nu \Sigma_f(\overline{r}) \phi^*(\overline{r}) dr. \tag{12}$$

The optimum real flux is constructed as in equation (8),

$$\phi_R^*(\overline{r}) = \sum_{i=1}^I a_i \phi_i(\overline{r}),$$

and the optimum poison concentrations are again determined by an equation analagous to equation (4). After invoking the eigenvalue relationships of equation (10), the poison concentrations are given by

$$\Sigma_p(\overline{r}) = \frac{\nu \Sigma_f(\overline{r}) \sum_{i=1}^I a_i (1-\lambda_i) \phi_i(\overline{r})}{\sum_{i=1}^I a_i \phi_i(\overline{r})}. \tag{13}$$

This method at last seems to handle the one-group problem rather well. Although it does not allow for control poisons in the reflector, that location is not popular for poisons (other than the soluable type) in the kind of reactors at which this analysis was aimed. Other sets of eigenfunctions are possible which allow for reflector control rods, but these bring with them the requirement for optimization searches which are not otherwise required in the one-group formulism. Such a set will be introduced and used by this author in the following chapters, as he extends the approach to multigroup formulism, where such searches apparently cannot be avoided.

CHAPTER III

THE POISON DISTRIBUTION PROBLEM IN MULTIGROUP FORMULISM

In contrast to some of the previous authors,[1-3] who used modified forms of two-group formulism in the hope of thereby obtaining direct solutions, this author has chosen to use the complete multigroup formulism. In principle, at least, fissions and births will be considered in all groups. The control poison will be generally considered to have a cross section in each group. Because of this general treatment, the resulting method should be applicable not only to the analysis of the current water moderated reactors, but also to fast reactors.

In the multigroup diffusion theory formulism, for K groups, the criticality equation becomes a set of K coupled equations, one for each energy group. Each of these equations has the general form

$$-\nabla \cdot D^k(\bar{r}) \nabla \phi^k(\bar{r}) + \Sigma_a^k(\bar{r}) \phi^k(\bar{r}) - \chi^k \sum_{n=1}^{K} \nu \Sigma_f^n(\bar{r}) \phi^n(\bar{r})$$

$$+ \Sigma_R^k(\bar{r}) \phi^k(\bar{r}) - \sum_{n=1}^{K} \Sigma_s(n \to k, \bar{r}) \phi^n(\bar{r}) + \Sigma_p^k(\bar{r}) \phi^k(\bar{r}) = 0. \quad (14)$$

The group fluxes, $\phi^k(r)$, are required to satisfy homogeneous boundary conditions and to be continuous, along with their net currents, $D^k(\bar{r}) \nabla \phi^k(\bar{r})$, at internal material interfaces.

The multigroup equations are not self-adjoint. The adjoint fluxes to the critical system satisfy a set of K equations of the form

$$-\nabla \cdot D^k(\overline{r}) \nabla \psi^k(\overline{r}) + \Sigma_a^k(\overline{r}) \psi^k(\overline{r}) - \nu \Sigma_f^k(\overline{r}) \sum_{n=1}^{K} \chi^n \psi^n(\overline{r})$$

$$+ \Sigma_R^k(\overline{r}) \psi^k(\overline{r}) - \sum_{n=1}^{K} \Sigma_s(k \rightarrow n, \overline{r}) \psi^n(\overline{r}) + \Sigma_p^k(\overline{r}) \psi^k(\overline{r}) = 0 \qquad (15)$$

and the adjoint boundary and interface conditions. Chapter IV will discuss some of the eigenvalue equations associated with equations (14) and (15). Orthogonality properties among the eigenfluxes and their adjoints will be demonstrated in that chapter for the sets of interest.

The power-flux relationship is given by

$$P(\overline{r}) = A \sum_{k=1}^{K} \Sigma_f^k(\overline{r}) \phi^k(\overline{r}). \qquad (16)$$

This equation contains the crux of the problem for the full treatment of the multigroup formulation. Unlike equation (2) for the one-group case, specification of an optimum power shape does not allow the solution for the multigroup fluxes. This occurs because there is no a priori optimum flux spectrum.

To understand the constraint on the energy dependence of the optimum flux, it is instructive to first proceed as if it were given. If this were true, each group flux could be written as the product of a known function and one of

the other fluxes. These relationships have the form

$$\phi^k(\bar{r}) = S^k(\bar{r})\phi^K(\bar{r}), \qquad \text{for } k = 1 \text{ to } (K-1). \qquad (17)$$

For convenience in notation, the identity relation for the group K flux may be used to define $S^K(\bar{r}) \equiv 1$. Equations (16) and (17) then form a set of K expressions in K unknown flux functions. The equations are linear, and the solutions for the optimum $\phi^k(\bar{r})$ would be straightforward, yielding

$$\phi^k(\bar{r}) = \frac{S^k(\bar{r})P*(\bar{r})}{A\sum_{n=1}^{K} S^n(\bar{r})\Sigma_f^n(\bar{r})}. \qquad (18)$$

(The fact that this could lead to a physically unrealizable flux, as in equation (3), Chapter II, is ignored here for the moment.) Then, upon substitution of these functions into equations (14), all terms would be known except the $\Sigma_p^k(r)$, for which solutions would again be straightforward. These solutions would have the form

$$\Sigma_p^k(\bar{r}) = \nabla \cdot D^k(\bar{r})\nabla\left[\frac{S^k(\bar{r})P*(\bar{r})}{\sum_{n=1}^{K} S^n(\bar{r})\Sigma_f^n(\bar{r})}\right]\frac{\sum_{n=1}^{K} S^n(\bar{r})\Sigma_f^n(\bar{r})}{S^k(r)P*(r)}$$

$$+ \chi^k \sum_{n=1}^{K} S^n(\bar{r})\nu\Sigma_f^n(\bar{r})/S^k(\bar{r}) - \Sigma_a^k(\bar{r}) - \Sigma_R^k(\bar{r})$$

$$+ \sum_{n=1}^{K} \Sigma_s(n{\to}k,\bar{r})S^n(\bar{r})/S^k(\bar{r}). \qquad (19)$$

The solution, then, for some arbitrary flux spectrum, would be a different specification of the macroscopic poison cross section function in each group, with no a priori relationship among them.

To achieve such a distribution of macroscopic poison cross sections physically, however, would be quite difficult. It is very unlikely that there exists a set of K poison materials, each with a finite cross section in only one of the energy groups, that could be distributed according to each of the equations (19). More likely, each position, $\bar{r}$, would require a different poison material, probably a mixture, with an energy dependence for its cross section which would match that specified by all of the equations (19) for point $\bar{r}$.

Except for some problems in breeder reactor design, where it is important to breeding, there is no desire to specify the neutron energy spectrum. Generally, however, it is desirable to use only one, pre-selected material for a control poison. In that case, the energy dependence of the absorption cross section is known, and the $\Sigma_p^k(\bar{r})$ of equations (19) must be constrained to

$$\Sigma_p^k(\bar{r}) = N(\bar{r})\sigma_p^k. \tag{20}$$

Thus, the real variables of the problem must be considered to be the $s^k(\bar{r})$ unknown spectrum functions, of which there are (K-1), and the $N(\bar{r})$ poison distribution function. This gives a total of K unknown functions.

The K relationships among these unknowns are derived by making the substitutions from equations (20) into equations (19).

$$N(\bar{r})\sigma_p^k = \nabla \cdot D^k(\bar{r}) \nabla \left[\frac{s^k(\bar{r})P*(\bar{r})}{A\sum_{n=1}^{K} s^n(\bar{r})\Sigma_f^n(\bar{r})} \right] \frac{A\sum_{n=1}^{K} s^n(\bar{r})\Sigma_f^n(\bar{r})}{s^k(\bar{r})P*(\bar{r})}$$

$$+ \chi^k \sum_{n=1}^{K} s^n(\bar{r})\nu\Sigma_f^n(\bar{r})/s^k(\bar{r}) - \Sigma_a^k(\bar{r}) - \Sigma_R^k(\bar{r})$$

$$+ \sum_{n=1}^{K} \Sigma_s(n\rightarrow k,\bar{r}) s^n(\bar{r})/s^k(\bar{r}). \tag{21}$$

The problem is thus fully specified, leaving no arbitrarily chosen variables. However, because the K equations (21) are nonlinear in the variable functions, the set of solutions, although finite, may contain more than one member. In fact, experience has shown there will be K solutions, all but one of which will involve some group fluxes with negative magnitudes. Although the physical realities of the problem allow the selection of the proper solution (the one with all positive group fluxes), these constraints may only be expressed mathematically as inequalities.

$$s^k(\bar{r}) > 0 \qquad \text{for all k.} \tag{22}$$

Unfortunately, such inequalities are not suitable for

substitution into equations (21) to analytically produce a linear set of equations. Consequently, equations (21) are not solved directly.

Considering now the problem that equations (16) and (18) will not be valid for arbitrary $P(\bar{r})$ and $\Sigma_f^k(\bar{r})$ functions, it is best to adopt a viewpoint somewhat different than that of Jackson and Chapter I. If $P^*(\bar{r})$ in equations (18) is assumed to be chosen such that it is the optimum function that satisfies equation (16), while allowing the fluxes to satisfy their continuity conditions, then it is just the function P_R^* sought by Jackson. Equations (18) through (21) are all valid if P^* is assumed to be this optimum realizable power distribution.

Thus, the complexity added to the problem, in going from one-group to multigroup theory, is that a nonlinearity has been introduced. In addition to determining P_R^* from P^*, equations (21) must also be solved.

Unfortunately, this author has been unable to find an eigenfunction set capable of providing an analytical solution to the coupled, nonlinear requirements of determining the optimum realizable power distribution and the associated distribution of a previously selected control material. However, it will be shown that proper choice of an eigenfunction expansion technique can lend considerable power to the search for a solution.

CHAPTER IV

SELECTION OF A MULTIGROUP EIGENFUNCTION SET

In order to find a useful set of eigenfunctions, it is first necessary to find some linear portion of the problem suitable for expansion. Because of their nonlinearity, eigenfunctions associated with equations (21) would be no easier to find than the direct solution of these equations. Thus, the "natural modes" or Kaplan[11] for this particular problem are impractical for expansion purposes because they also are given by a nonlinear equation.

The flux portion of the problem, as given by equations (14), is linear in the unknown fluxes, $\phi^k(\bar{r})$. It is non-linear overall only due to the control terms, $\Sigma_p^k(\bar{r})\phi^k(\bar{r})$, because of the unknown distributional function in the poison cross sections. Eigenfunctions to expressions associated with equations (14) can thus readily be found if the control terms are left out of the eigenvalue equations.

Of the several possible sets of multigroup flux eigen-functions, some are more useful than others. As was the case in the one-group formulism, the "natural modes" to the unpoisoned flux problem contain more information than is needed, or wanted, about the flux in unfueled portions of the reactor. As will be demonstrated, the "fission eigen-functions" of Jackson,[7] on the other hand, contain a

18

degeneracy in the energy variable, and are suitable only for expansions of the spatial dimension. Other eigenfunctions will be introduced which overcome both of these objections. Before this is done, however, the author will introduce the multigroup, finite difference formulism that must generally be employed to calculate whatever eigenfunctions are to be used. This will allow the comparison of the various eigenfunction sets within the actual context of their computation and use, helping to more clearly exhibit their various strengths and failings.

The finite difference approximation to the spatial dimension is quite analogous to the multigroup treatment of the energy dimension. A finite number of points, $\bar{r}_j$, are chosen along the ordinates of the reactor, and the continuous functions of flux, power, etc., are represented by their values at these points. Thus, if J points are chosen, an unknown function becomes a set of J unknown values, usually written together as a vector. Applying this to the group fluxes of equations (14), the flux in group k would be represented as

$$\phi^k(\bar{r}) = \{\phi^k(\bar{r}=\bar{r}_1); \phi^k(\bar{r}=\bar{r}_2); \ldots; \phi^k(\bar{r}=\bar{r}_J)\}. \tag{23}$$

In order to simplify notation for the finite difference equations, the following convention will be used. Any function, f, of energy and space will have its value at point j in energy group k denoted as f(k,j).

Equations (14) may be rewritten for each spatial position, $\bar{r}(j)$, for which a flux value is to be determined. The second derivatives associated with the gradient terms may be approximated by the second central difference method.[16]   For example, in one-dimensional, slab geometry, the gradient term becomes

$$-\nabla \cdot D^k(\bar{r})\nabla \cdot \dot{\phi}^k(\bar{r}) = -\frac{d}{dx}\{D^k(x)\frac{d}{dx}\phi^k(x)\} \tag{24}$$

which, for constant diffusion coefficient and evenly spaced $x(j)$, would be approximated as

$$-D^k\frac{d^2\phi^k(x)}{dx^2} \simeq -D(k)[\phi(k,j+1)-2\phi(k,j)+\phi(k,j-1)]/\Delta x^2. \tag{25}$$

From this it can be seen that in one dimension the set of J finite difference equations, replacing the kth multigroup equation (14), each contain three of the flux values in the kth group and are therefore coupled spatially. Thus the finite difference representation of the criticality conditions will consist of $J \cdot K$ equations of the general form

$$C(k,j,j+1)\phi(k,j+1)+C(k,j,j)\phi(k,j)$$

$$+C(k,j,j-1)\phi(k,j-1)+\Sigma_a(k,j)\phi(k,j)$$

$$-\sum_{n=1}^{K}\Sigma_s(n\rightarrow k,j)\phi(n,j)+\Sigma_p(k,j)\phi(k,j)$$

$$-X(k)\sum_{n=1}^{K}\nu\Sigma_f(n,j)\phi(n,j)+\Sigma_R(k,j)\phi(k,j) = 0, \tag{26}$$

where the C's will depend upon choice of geometry, spatial

variation of the diffusion coefficient, and spacing of the
evaluation points, $\bar{r}(j)$. In analogy to equation (20), the
finite difference representation of the previously chosen
control material is

$$\Sigma_p(k,j) = N(j)\sigma_p(k), \tag{27}$$

where the $\sigma_p(k)$ is a set of known values. Thus, the finite
difference, multigroup approximation to the criticality
condition, as represented by equations (26), has $J \cdot (K-1)$
unknowns: the $J \cdot K$ fluxes, $\phi(k,j)$, which are coupled in
energy and space, and the J poison concentrations, $N(j)$,
which are coupled to the fluxes of point j at all energies.

The real flux in the reactor is now represented by
only $J \cdot K$ values, or as a vector with that many elements.
An arbitrary flux vector will thus have $J \cdot K$ degrees of
freedom, and, if it is to be represented by a sum of known
vectors, this sum must also have that many degrees of free-
dom in the unknown coefficients. Stated in terms of vector
analysis, a set of $J \cdot K$ linearly independent vectors is
necessary to span the space of all vectors with $J \cdot K$ ele-
ments.[16,17] Therefore, in order to expand the multi-
group flux vector with a set of eigenvectors, it is neces-
sary that there be enough of them to span the desired vector
space. If only H of the spatial points are of interest to
the power specification problem, then only $H \cdot K$ flux values
are of interest. This $H \cdot K$ subspace of the general $J \cdot K$

vector space may be spanned by only $H \cdot K$ of the vectors,
provided a set is chosen that is still linearly independent
when only the $H \cdot K$ elements of interest are considered. This
is in contrast to the continuous function formulism, where
an infinite number of functions, forming a complete set
over some space, are required to fully construct an arbi-
trary function over that space. Because in the finite
difference case the physical representation of the problem
was truncated at a finite number of points and energies,
a finite number of vectors is sufficient to construct any
vector representing the flux values at each point and energy.

If the control terms of equations (26) are neglected,
and the remaining terms are rearranged into various eigen-
value problems, the resulting $J \cdot K$ equations will be linear
in the $J \cdot K$ unknown fluxes. The sets of solutions to these
problems may be found by well established methods,[18,19]
which will be discussed in the next chapter. These problems
may all be written in matrix notation in the form

$$[A]\phi_i = \alpha_i [B]\phi_i, \tag{28}$$

where α_i is the generalized eigenvalue. The actual form of
the matrix operators, [A] and [B], will depend upon the
choice of eigenvalue problem. These matrices will be square,
of order $(J \cdot K)$, and the eigenvectors, ϕ_i, will have $(J \cdot K)$
elements.

Before considering particular eigenvalue problems,
some observations about the general formulation are in order.

The adjoint to equation (28) is given by

$$[A]^T \psi_i = \alpha_i [B]^T \psi_i. \qquad (29)$$

Following Hansen and Clark,[20] the orthogonality relationship may be derived as follows. Assuming that [A] has an inverse, then the "solution" to equation (28) may be written as

$$\phi_i / \alpha_i = [A]^{-1} [B] \phi_i, \qquad (30)$$

and the "solution" for the adjoint problem is

$$\psi_i / \alpha_i = \{[A]^T\}^{-1} [B]^T \psi_i = \{[A]^{-1}\}^T [B]^T \psi_i. \qquad (31)$$

The adjoint to equation (30) is

$$\theta_i / \alpha_i = \{[A]^{-1} [B]\}^T \theta_i = [B]^T \{[A]^{-1}\}^T \theta_i. \qquad (32)$$

Since matrix multiplication is not commutative, in general, the adjoint to the "solution" need not be the same as the "solution" to the adjoint equation. Thus, in the terminology of Hansen and Clark,[20] the operations of adjoining and "solving" are not generally commutative. Use of this fact may be made in selecting a set of adjoint vectors with desirable orthogonality conditions. The θ's satisfy the usual, unweighted condition $\theta_i^T \phi_j = \delta(i,j)$, and are thus orthogonal only over the whole J·K vector space. A more useful orthogonality condition may be derived for the ψ_j adjoint set. If the operator [B] is applied to both sides of equation (30), the result is

$$[B]\phi_i/\alpha_i = [B][A]^{-1}[B]\phi_i. \tag{33}$$

Now, if $[B]\phi_i$ is considered as a vector, its adjoint, denoted by $\bar{w}_i$, must satisfy the equation

$$\bar{w}_i/\alpha_i = \{[A]^{-1}\}^T [B]^T \bar{w}_i. \tag{34}$$

By comparison of equations (31) and (34), it can be seen that $\bar{w}_i$ is identical to ψ_i. By taking the inner product of the transpose of equation (34) with the vector $[B]\phi_j$, and subtracting the inner product of $\bar{w}_i^T$ with equation (33), the result is

$$\langle \bar{w}_i/\alpha_i ; [B]\phi_j \rangle - \langle \bar{w}_i ; [B]\phi_j/\alpha_j \rangle$$

$$= \bar{w}_i^T [B][A]^{-1}[B]\phi_j - \bar{w}_i^T [B][A]^{-1}[B]\phi_i = 0. \tag{35}$$

With the scalar α's removed from the inner products, this becomes

$$(1/\alpha_i - 1/\alpha_j)\bar{w}_i^T [B]\phi_j = 0. \tag{36}$$

Substituting ψ_i for $\bar{w}_i$, this gives the orthogonality condition

$$\psi_i^T [B]\phi_j = \delta(i,j). \tag{37}$$

Another property of interest in selecting a set of eigenvectors is the number of eigenvectors associated with a particular formulation. This can be found by considering the determinate of $[A]-\alpha[B]$ as a function of α. In order to satisfy equation (28), α must satisfy

$$\det\{[A] - \alpha[B]\} = 0. \tag{38}$$

In general, expanded analytically, equation (38) results in an equation in α of degree equal to the order of the square matrices [A] and [B]. The coefficient of the α^n term will be equal to the sum of the determinates of all possible matrices formed by replacing n columns (or rows) of the [A] matrix with their corresponding columns (or rows) of the [B] matrix. It has already been specified that [A] must have an inverse, so det[A] is not zero. However, no such restriction is placed on [B]. The rank of a matrix is equal to the number of rows (or columns) in the matrix that are linearly independent. If [B] is of rank m, then all matrices formed by replacing rows of [A] by more than m corresponding rows of [B] must contain linearly dependent rows, and, therefore, have zero determinates. The coefficients of all terms α^n, where n>m, will therefore be zero. Equation (38) will thus be an m degree equation in α and have no more than m solutions. The number of eigenvalues (and eigenvectors) to equation (38) will therefore be limited to being less than or equal to the rank of the matrix [B].

Except for the "ω modes" to the Laplace transformed kinetics problem, where they indicate oscillatory behavior in time, complex or imaginary solutions for α seem to have no convenient physical interpretation. In practice, however, such roots do not seem to occur for formulations where the eigenvalue may be considered as the multiplier of

some real physical parameter. Double or higher order re-
peated real roots may occur in special cases of some of
the eigenvalue formulations to be considered herein. In
such cases, it is not always possible to find a complete
set of eigenvectors. If not, methods exist for treating
the problem by supplementing the eigenvectors with a set
of "principal vectors".[16] Such special cases will not
be considered further in this paper. Hereafter, it will
be assumed that, for matrix [B] with rank m, only the cases
for which there are m true eigenvectors are considered.

Considering now the flux "natural modes" associated
with the criticality equations (28) with the control terms
removed, the individual equations are of the form

$$C(k,j,j+1)\phi_i(k,j+1)+C(k,j,j)\phi_i(k,j)$$

$$+C(k,j,j-1)\phi_i(k,j-1)+\Sigma_a(k,j)\phi_i(k,j)$$

$$-X(k)\sum_{n=1}^{K}\nu\Sigma_f(n,j)\phi_i(n,j)+\Sigma_R(k,j)\phi_i(k,j)$$

$$-\sum_{n=1}^{K}\Sigma_s(n\rightarrow k,j)\phi_i(n,j) = \gamma_i\phi_i(k,j),\qquad(39)$$

where γ is the eigenvalue for these modes. In matrix no-
tation, this is

$$[A]\phi_i = \gamma_i[I]\phi_i.\qquad(40)$$

All of the materials parameters, with the exceptions of the
control poison, are included in matrix [A], which will have

an inverse unless the reactor is just critical. (That is, det[A] = 0 is just the criticality condition.) The identity matrix, [I], is its own inverse. Thus, for a J point, K energy group representation of the reactor, there will be J·K eigenvalues and eigenvectors satisfying the orthogonality conditions

$$\psi_j^T [I] \phi_i = \psi_j^T \phi_i = \delta(i,j). \tag{41}$$

These J·K linearly independent eigenvectors (assuming no degeneracy of roots in [A]) are sufficient to expand any arbitrary flux vector of the chosen mathematical model. Still, however, these J·K vectors may contain more information than is of interest or more than can conveniently be used. If only H of the J spatial points lie within fueled regions, where the power is finite, then only H·K flux values need be specified. It would be convenient if the rest could be neglected. However, these eigenvectors, with J·K elements, satisfy the orthogonality conditions with their adjoints only over the entire J·K range. Thus, J·K possible variables (the expansion coefficients for all the eigenvectors) have been created to handle a case where there are at most H·K degrees of freedom of any interest.

Experience with the one-group problem suggests the selection of the "fission modes" as a set of eigenvectors that would be complete over the fueled region alone. The individual equations to this formulation are

$$C(k,j,j+1)\phi_i(k,j+1)+C(k,j,j)\phi_i(k,j)$$

$$+C(k,j,j-1)\phi_i(k,j-1)+\Sigma_a(k,j)\phi_i(k,j)$$

$$-\sum_{n=1}^{K}\Sigma_s(n\rightarrow k,j)\phi_i(n,j)+\Sigma_R(k,j)\phi_i(k,j)$$

$$=\lambda_i X(k)\sum_{n=1}^{K}\nu\Sigma_f(n,j)\phi(n,j), \tag{42}$$

where λ_i, the fission mode eigenvalue, is equal to the inverse effective reactivity of the mode i. The ortho-gonality condition leads to an expression for the expansion coefficients for a given flux vector, ϕ, given by

$$a_i = \psi_i^T[B]\phi, \tag{43}$$

that weights any fluxes in unfueled portions of the reactor by zero. Just as in the one-group case, only the fluxes in the fueled regions need to specified to determine the expansion coefficients. However, the [B] matrix to this problem will have J sets of linearly dependent rows, K rows per set. This is due to a physical degeneracy in the energy dimension of the fission process: the energy spectrum of the neutrons born in fission is virtually independent of the energy spectrum of the neutrons causing the fissions. Thus, as can be seen by the right hand side of equation (42), the births due to all fissions at each point are summed and then distributed over the energy groups at the same point by the function $X(k)$. Physically speaking, the left

hand sides of equations (42) are being "driven" by a neu-
tron source with a constant spectrum, $X(E)$, and this
driving source spectrum is independent of the flux spectrum.
The resulting [B] matrix has the form shown in Figure (1).
As can be seen in the figure, the K rows associated with
the K equations for a given point will differ only by
factors of $X(k)$. Thus, all K of the rows for that point
will be linearly dependent. The rows for different space
points are not linearly dependent, however. The rank of
matrix [B] for the "fission modes" is then J, the number of
space points. This set of eigenvectors, although orthogonal
over the proper region of the reactor, is not complete. The
J vectors are insufficient to expand the H·J elements of
interest in the flux vector.

A set of eigenfunctions that is complete over the de-
sired space can be generated by equations of the form

$$C(k,j,j+1)\phi_i(k,j+1)+C(k,j,j)\phi_i(k,j)$$

$$.+C(k,j,j-1)\phi_i(k,j-1)+\Sigma_a(k,j)\phi_i(k,j)$$

$$-X(k)\sum_{n=1}^{K}\nu\Sigma_f(n,j)\phi_i(n,j)+\Sigma_R(k,j)\phi_i(k,j)$$

$$-\sum_{n=1}^{K}\Sigma_s(n{\rightarrow}k,j)\phi_i(n,j) = -\beta_i\sigma_p(k,j)\phi_i(k,j), \qquad (44)$$

where $\sigma_p(k,j)$ is the poison cross section in the kth group
(for some unit density of poison) for points where control
poison is to be used, and is zero where no control poison

$$
\begin{bmatrix}
X(1)\nu\Sigma_f(1,1) & X(1)\nu\Sigma_f(2,1) & \cdots & 0 & X(1)\nu\Sigma_f(1,2) & X(1)\nu\Sigma_f(2,2) & \cdots & 0 & \cdots \\
X(2)\nu\Sigma_f(1,1) & X(2)\nu\Sigma_f(2,1) & \cdots & 0 & X(2)\nu\Sigma_f(1,2) & X(2)\nu\Sigma_f(2,2) & \cdots & 0 & \cdots \\
\cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdots & \cdot & \cdots \\
0 & 0 & \cdots & X(1)\nu\Sigma_f(1,2) & \cdots & X(1)\nu\Sigma_f(2,2) & \cdots \\
0 & 0 & \cdots & X(2)\nu\Sigma_f(1,2) & \cdots & X(2)\nu\Sigma_f(2,2) & \cdots \\
\cdot & \cdot & \cdots & \cdot & \cdots & \cdot & \cdots
\end{bmatrix}
\begin{bmatrix}
\phi(1,1) \\ \phi(2,1) \\ \cdot \\ \phi(1,2) \\ \phi(2,2) \\ \cdot
\end{bmatrix}
$$

Figure 1   The [B] Matrix and Flux Vector for the "Fission Eigenfunctions"

The rows of the matrix represent equations for the fluxes in the first and second energy groups at the first point and the first and second energy groups at the second point, respectively. The columns (and thus the fluxes) correspond to the first and second energy groups at the first point and the first and second energy groups at the second point, respectively.

is to be used. Physically, this is equivalent to specifying
that the concentration of poison is the same at each con-
trol point. The eigenvalue, β_i, varies the magnitude of
all the concentrations to achieve criticality in each mode.
In this case, which the author choses to call the "poison
eigenfunctions", matrix [B] will be diagonal, with zeros
on the diagonal for all the rows corresponding to points
that have no control poison. For a problem with G control
points, matrix [B] will be of rank G·K, and there will be
G·K eigenvectors which are complete in energy and space
only for the regions of the reactor chosen for control. In
matrix form, equations (44) are written as

$$[A]\phi_i = -\beta_i[\sigma_p]\phi_i, \tag{45}$$

where $[\sigma_p]$ is the [B] matrix for the poison eigenvectors.

These eigenvectors still have J·K elements, and so re-
present the flux at the noncontrol points as well. The
criticality condition limits the flux at noncontrol points
to values which are representable by sums of the poison
eigenvectors. The proof that the poison eigenvectors span
the "space" of critical configurations is quite straight-
forward. Consider the finite difference criticality equa-
tion for the controlled reactor,

$$[A]\phi = -[\Sigma_p]\phi. \tag{46}$$

The matrix $[\Sigma_p]$ is diagonal and represents any critical

poison distribution, generally with different concentra-
tions at each control point. Like $[\sigma_p]$, this critical con-
trol matrix, $[\Sigma_p]$, will have zeros on the diagonal for all
noncontrol points. Assuming that a complete set of poison
eigenvectors is available, the expansion coefficients for
the critical flux may be found from the orthogonality rela-
tions.

$$a_i = \psi_i^T [\sigma_p] \phi \tag{47}$$

The flux at the control points, at least, is represented
by

$$\phi^* = \sum_{i=1}^{I} a_i \phi_i . \tag{48}$$

Now, if ϕ^* is different from ϕ, it must differ only at non-
control points, since the poison eigenvectors are complete
over the control points. Let any difference be denoted as

$$\theta = \phi - \phi^* . \tag{49}$$

It is easily seen that the constructed flux satisfies the
criticality equation (46) by using the eigenvalue relations
(45).

$$[A]\phi^* = [A] \sum_{i=1}^{I} a_i \phi_i = -[\sigma_p] \sum_{i=1}^{I} a_i \beta_i \phi_i \tag{50}$$

Since the matrices $[\sigma_p]$ and $[\Sigma_p]$ have zeros on the diagonal
for all noncontrol points, the substitution of ϕ^* in the
criticality equation yields a relationship dependent only

upon the fluxes at control points, where ϕ^* and ϕ are identical.

$$-[\sigma_p] \sum_{i=1}^{I} a_i \beta_i \phi_i = -[\Sigma_p] \sum_{i=1}^{I} a_i \phi_i \tag{51}$$

Thus, ϕ^* also satisfies the criticality relation. Now, by substituting $\phi^*-\phi$ for ϕ in equation (46),

$$[A](\phi^*-\theta) = -[\Sigma_p](\phi^*-\theta). \tag{52}$$

Since ϕ^* already satisfies equation (46), this may be reduced to

$$[A]\theta = -[\Sigma_p]\theta. \tag{53}$$

However, θ must be zero at all control points, and the operator $[\Sigma_p]$ multiplies all noncontrol points by zero, so

$$[\Sigma_p]\theta \equiv \overline{0}. \tag{54}$$

Thus, if the difference vector, θ, exists, it must satisfy

$$[A]\theta \equiv \overline{0}. \tag{55}$$

So, for a nontrivial solution for θ to exist, it is necessary that

$$\det[A] \equiv 0. \tag{56}$$

If this condition is not satisfied, then θ is the zero vector and the expanded flux is identical to the actual flux. If equation (56) is satisfied, the uncontrolled reactor must be critical in some mode, i. For this case, however,

a poison eigenvector will exist with a zero poison eigen-value.

$$[A]\phi_i = -0[\sigma_p]\phi_i \equiv \overline{0} \tag{57}$$

Thus, if it existed, θ must be identical to ϕ_i, and thus already included in the expansion of the flux, in which case it could not be the difference between the expanded and the true flux. Thus, in all cases where there is a complete set of poison eigenvectors, they are capable of representing any critical reactor configuration attainable by distributing poison at the designated control points.

These poison eigenfunctions at last seem to be able to adequately represent the physical situation. For a reactor with G control points (e.g., rods, regions, etc.), there are only G degrees of freedom for control purposes. Another $G \cdot (K-1)$ variables are needed for expansion, however, because the spectrum is not known a priori. This set of eigenvectors provides the minimum number of necessary variables and insures that they are complete over only those spatial points that may actually be directly controlled.

CHAPTER V

GENERATION OF THE POISON ENGENVECTORS

Before the eigenvectors to a particular problem can be
used in the solution of that problem, the eigenvectors them-
selves must be obtained. Most computer codes available today
for the solution of the diffusion equations are concerned
only with the fundamental mode of the fission eigenfunctions.
That is, they find the eigenvalue that is the inverse of the
reactivity for the real flux shape in the reactor. Such
codes are designed to find only the one mode for only the
one eigenvalue formulation, which is degenerate in the energy
dimension. Thus, there is little hope for the modification
of this type of code to find the complete set of poison
eigenfunction.

An exception is the MULE code,[14] which is capable of
finding any desired mode for three types of two-group prob-
lems, the fission eigenfunctions, the poison eigenfunctions
for a thermal poison only, and the time or kinetic eigen-
functions, with or without delayed neutrons. The fission
modes satisfy the equation

$$[L]\phi_i = \lambda_i [S]\phi_i,$$ (58)

where [L] contains all of the leakage and absorption loss

35

terms and [S] contains the fission source terms. As dis-
cussed in Chapter IV, the [S] matrix has a rank equal to
the number of spatial points, J, and thus equation (58)
produces only J eigenvectors. The thermal poison eigen-
functions have a similar degeneracy. They satisfy the
relation

$$\{[L] - \frac{1}{K_{eff}}[S]\}\phi_i = \mu_i[Q]\phi_i, \tag{59}$$

where [Q] is a diagonal matrix with zeros on the diagonal
for all the fast group equations and Σ_p on the diagonal for
all the thermal group equations. Thus, the rank of [Q] is
also J, and this set of eigenvectors is also insufficient to
fully expand the two-group problem. The time modes, without
delayed neutron groups, satisfy the equation

$$\{[L] - \frac{1}{K_{eff}}[S]\}\phi_i = -\omega_i[V]\phi_i, \tag{60}$$

where [V] is a diagonal matrix with the inverse of the neu-
tron velocity for the appropriate group on each diagonal
location. The rank of [V] is thus 2J, and the eigenfunc-
tions of this type will be complete over energy and space.
This problem is very similar to the two-group poison eigen-
functions desired. If the poison fast and thermal cross
sections were substituted for the inverse velocities in the
fast and thermal groups, respectively, then true two-group
poison eigenfunctions could be found. However, as written,
the MULE code would impose the unwanted restriction that all
points would be control points. This problem could easily

be overcome by writing an additional input routine to allow the desired flexibility in control point locations.

Instead of altering MULE, this author has chosen to write a new eigenvector generator that can be run on a smaller computer, allowing the programmer-operator to have more convenient interaction with the program while calculations are in progress. The desirability of such an arrangement will become apparent shortly.

The actual calculation of an eigenvector is typically accomplished by the power iteration method.[18]   For the general eigenvalue problem of equation (28),

$$[A]\phi_i = \alpha_i[B]\phi_i, \tag{28}$$

there will be a set of I solutions, where I is less than or equal to the order of matrices [A] and [B]. These vectors can be combined in any arbitrary manner to form a vector

$$\bar{V} = \sum_{i=1}^{I} a_i\phi_i. \tag{61}$$

If [A] has an inverse, then the eigenvalue problem may be written in the form of equation (30).

$$\phi_i/\alpha_i = [A]^{-1}[B]\phi_i \tag{30}$$

The effect of operating on vector $\bar{V}$ by the matrix $[A]^{-1}[B]$ can be found by substitution for each eigenvector from equation (30).

$$[A]^{-1}[B] \sum_{i=1}^{I} a_i \phi_i = \sum_{i=1}^{I} a_i \phi_i / \alpha_i \qquad (62)$$

The vector on the right hand side of equation (62) has thus had its component of the ith eigenvector changed by the factor $1/\alpha_i$. If α_i is greater than one, the contribution from vector i will be decreased; if it is less than one, the contribution will be increased. Therefore, by repeatedly operating on any vector $\bar{V}$ by the matrix $[A]^{-1}[B]$, the eigen-vector component with the smallest absolute value of α will be the one to grow the fastest or decrease the slowest in absolute value. If the new vector is renormalized after each successive operation, then eventually, only this domin-ant eigenvector will remain. As can be seen from equation (30), when the guess vector has converged to the eigenvector ϕ_i, the renormalization factor will be just the inverse of the associated eigenvalue. How fast the other vectors die away is governed by the dominance ratio. If vector i has the smallest value of α, then the ratio of increase in the contribution of vector i, relative to vector j, between iterations is given from equation (62) as

$$\left| \frac{a_i \phi_i / \alpha_i}{a_i \phi_i} \right| \Bigg/ \left| \frac{a_j \phi_j / \alpha_j}{a_j \phi_j} \right| = \left| \frac{\alpha_j}{\alpha_i} \right| . \qquad (63)$$

The greater the dominance ratio, the more rapidly vector j will die away compared to vector i.

Obviously, if matrix [B] also has an inverse, it is possible to find the largest eigenvalue and associated eigenvector by writing the problem as

$$\alpha_i \phi_i = [B]^{-1}[A]\phi_i, \tag{64}$$

and operating repeatedly with the matrix $[B]^{-1}[A]$. Still, the vectors with eigenvalues of intermediate modulus must be found by some other method. There are two possible methods. The first is successive elimination[18] of the roots as they are found. This can be done by choosing a starting vector that is orthogonal to all previously found eigenvectors, so that the sum in equation (61) no longer contains these components, or by modifying the original matrix[18] to eliminate known roots. This leaves the next root dominant. There are several disadvantages to such methods. The dominance ratios for the last modes are usually not very great, causing many iterations to be necessary for convergence to the next mode. Accuracy of the later modes also suffers because the calculations depend upon the results for all previous modes, and the errors accumulate.

A better method, proposed by Wielandt,[18,19] is to transform the eigenvalue problem to one in which the desired root is dominant. If a guess is made for the next value of α, then the eigenvalue problem may be rewritten as

$$\{[A] - \alpha_g[B]\}\phi_i = \delta_i[B]\phi_i, \tag{65}$$

where α_g is the value of the guess, and the eigenvalues of the transformed equation (64) are related to the original eigenvalues by

$$\alpha_i = \delta_i + \alpha_g. \tag{66}$$

The dominance ratio for the new equation is calculated using the δ's. Obviously, the smallest δ will be the one associated with the α_i closest to the α_g, and the closer the guess, the faster will be the convergence. In order to find any eigenvector, it is therefore only necessary to provide a guess at the eigenvalue which is closer to that eigenvalue than to any other, and to repeatedly multiply some starting vector by the matrix $\{[A]-\alpha_g[B]\}^{-1}[B]$ until convergence is obtained. Of course, standard acceleration technique may be used to speed the convergence of the vector. The Wielandt transformation itself may be used as an acceleration technique if the eigenvalue guess is updated using partially converged values of δ.[21,22]

To aid in guessing close to the eigenvalues, it is helpful to consider the physical significance of the eigenvalue problem chosen. The poison eigenfunctions satisfy equation (45).

$$[A]\phi_i = -\beta_i[\sigma_p]\phi_i \tag{45}$$

The [A] matrix contains all of the terms for the uncontrolled reactor, and matrix $[\sigma_p]$ contains the terms for absorption

in unit concentrations of poison placed at all control points.

The eigenvalue serves the function of uniformly varying the

amount of poison at all the control points. If the reactor

is supercritical without control poison, then the funda-

mental mode (the most reactive one with flux that is every-

where positive) will require some control poison insertion

to reach the critical state. Thus, the associated β_1 will

be positive. A good estimate of its value can be made from

first order perturbation theory, if the fundamental fission

mode and its adjoint have already been found by one of the

standard diffusion codes. This is usually not necessary,

however.

In order to understand how the rest of the modes be-

have, it is instructive to examine a simple example. For

a two-group representation of a bare, homogeneous cylindri-

cal reactor of infinite axial extent, the equations for the

two-group fluxes in the poison eigenvalue formulation are

given by

$$-D(F)\nabla^2\phi_i(F,r)+\{\Sigma_a(F)+\Sigma_R(F)-\nu\Sigma_f(F)\}\phi_i(F,r)$$

$$-\nu\Sigma_f(T)\phi_i(T,r) = -\beta_i\sigma_p(F)\phi_i(F,r) \tag{67a}$$

for the fast group, and

$$-D(T)\nabla^2\phi_i(T,r)+\Sigma_a(T)\phi_i(T,r)-\Sigma_R(F)\phi_i(F,r)$$

$$= -\beta_i\sigma_p(T)\phi_i(T,r) \tag{67b}$$

for the thermal group. Because the energy and space variables

are separable for this example, the solutions to equations
(67) may be written as fluxes of the form

$$\phi_i(F,r) = S_i J_o(n_i r/R) \tag{68a}$$

$$\phi_i(T,r) = J_o(n_i r/R), \tag{68b}$$

where R is the radius of the reactor, J_o is the zeroth order
Bessel function, n_i is the ith zero of J_o, and S_i is the
fast to thermal flux ratio, which is a constant over the
entire reactor volume for each mode. Substitution of equa-
tions (68) into equation (67) allows the solution for the
eigenvalue, β_i, and the spectral constant, S_i, for each
mode. These solutions are

$$S_i = -\frac{b}{2a} \pm \sqrt{\frac{b}{2a}^2 - \frac{c}{a}} \tag{69}$$

$$\beta_i = \{S_i \Sigma_R(F) - D(T)[n_i/R]^2 - \Sigma_a(T)\}/\sigma_p(T), \tag{70}$$

where $a = \sigma_p(F) \Sigma_R(F)$

$$b = \sigma_p(T)\{D(F)[n_i/R]^2 + \Sigma_a(F) + \Sigma_R(F)\}$$

$$-\sigma_p(F)\{D(T)[n_i/R]^2 + \Sigma_a(T)\}$$

$$c = \nu\Sigma_f(T)\sigma_p(T).$$

The quadratic nature of the solution is due to the two-group
formulism. In general, the degree of the equation for S_i
will be equal to the number of energy groups.

The J_o function is a well known oscillatory function
which crosses the axis at the tabulated values, n_i.

By scaling its argument as $(n_i r/R)$, it is made to have zero
value at the extrapolated reactor boundary, R, and to pass
through zero (i-1) times within the reactor. For a parti-
cular choice of the index, i, the fast and thermal fluxes
will have the same spatial shape, differing only in magnitude
by the factor S_i. Because there are two groups, there are
two values of β and two values of S associated with each
of these shapes. From equation (69), it may be seen that
one value of S_i will always be positive, and the other
always negative. By equation (70), the negative S_i will
always be associated with a negative β_i, while the positive
S_i may have either a positive or negative value of β_i. In
fact, as discussed previously, when the uncontrolled reactor
is supercritical, the fundamental mode must have a positive
value of β. As the index, i, and thus n_i, is increased in
value, the leakage terms in equations (67), given by

$$-D(k)\nabla^2 J_o(n_i r/R) = D(k)[n_i r/R]^2 J_o(n_i r/R), \qquad (71)$$

will become more positive. In order to satisfy equations
(67), then, the values of β must become more negative. The
same conclusion may also be reached by physical reasoning.
The innermost region of any higher flux mode, where
$0 \leq r \leq n_i R/n_i$, is physically equivalent to the fundamental mode
of a smaller reactor. In order to make this smaller reac-
tor critical, less control poison will be needed because
of the greater neutron leakage. Thus, again, as n_i increases,
β_i must decrease.

For this case, then, the β's will occur in two sequences, denoted by β_i^+ and β_i^- for the values associated with the positive and negative values of S_i, respectively. These sequences satisfy the relations

$$\beta_{i+1}^+ < \beta_i^+ \tag{72a}$$

$$\beta_{i+1}^- < \beta_i^- < 0. \tag{72b}$$

From the shape of the fluxes, the index, i, may be derived by counting the number of times the flux passes through zero and adding one.

In general, the flux shapes of reactors with more complicated materials configurations will not be given by the analytic function, J_o. Over their region of completeness, they will still behave in a manner similar to that described in the last paragraph, however. Experience has shown that, if the flux values associated with noncontrol points are ignored, the remaining values will show the proper number of sign changes in both the energy and space variables. This is of great utility to the person wishing to generate a set of eigenvectors using the Wielandt transformation technique. Any guess at an eigenvalue will result in convergence to some eigenvalue and vector. By using the above criteria, it may easily be determined which eigenvalue has been found, and where the other eigenvalues must be located in relation to this and previously found eigenvalues. Thus, a few educated guesses, or, at worst, a few shots in the dark,

for values of β will result in enough information about the magnitude and spacing of the β sequences so that guesses near the remaining roots may be made quite readily.

While the evaluations and decisions associated with this initial searching and later prediction for values of β may easily be handled by the human mind, programming of the general process would be quite complicated and require much computer storage. Such complexities as the possible overlap of the β sequences, variability of the control (completeness) regions between specific problems, and the possibility of finding the same root more than once, greatly complicate the logic and bookkeeping necessary for a computer program that would be capable of finding the complete set of eigenfunctions with no operator interaction.

For these reasons, and for the purpose of illustration, the author chose to write a sample eigenvector generator designed to run on the IBM 1800 computer system. Briefly, the program reads the problem geometry from punched cards, sets up the iteration matrix, and asks, via the typewriter, that a guess be made for β. When this value is entered, along with the maximum number of iterations to be allowed on this guess, the program performs the iterations until convergence of the vector or the iteration limit is reached. The revised eigenvalue, the vector and, if convergence is obtained, the adjoint vector are then displayed to the operator on the typewriter, and the operator is given the option of having this output punched on cards for later use.

The program then cycles back and asks for another eigenvalue estimate. Through the use of sense switches, the operator also has the options of varying the convergence criteria according to the estimated numerical stability of the various modes, looking at any of the intermediate vectors in the iteration process, or specifying that the adjoint be found regardless of eigenvector convergence.

This program, named CODE2 by the author, is described more fully in Appendix I, and a listing appears as Appendix II. Because the program was written only for purposes of demonstration, it is much less general and sophisticated than it is allowed to be in theory. It is limited to two-group, one-dimensional calculations in cylindrical geometry. Extentions to more groups, more dimensions and other geometries are straightforward, following established techniques employed by typical modern diffusion codes. For the purpose of demonstrating the concepts and techniques introduced in this paper, however, the program is sufficient as written.

CHAPTER VI

THE MULTIGROUP OPTIMIZATION PROBLEM

As pointed out in Chapter III, the problem of finding
the optimum multigroup flux functions, and the distribution
of a control poison necessary to maintain these flux shapes
in the critical state, results in a set of nonlinear equa-
tions. There is no general analytical approach for solving
nonlinear systems, and the author has been unable to find a
specialized technique capable of providing a solution for
this particular problem. Therefore, this author has chosen
the numerical optimization approach to provide individual
solutions for individual reactors.

Numerical search techniques, in general, find solutions
for particular problems based on the results of sampling
calculations performed with various sets of the independent
variables. As applied to the problem of finding the optimum
poison distribution in a particular reactor, these variables
are the poison concentrations at each control point. An
initial set of poison concentrations is specified at the
beginning, and the resulting power distribution is calculated.
Comparison of this power distribution with the optimum dis-
tribution gives the "error" in power resulting from the use
of this poison distribution. By systematically varying the
individual poison concentrations and recalculating the power   -

distribution, the derivatives of the error are calculated as functions of the poison concentrations at each control point. Various algorithms may then be employed to predict the location of, or direction to, the poison distribution yielding the optimum power distribution. This results in a new poison distribution that is closer to the optimum, about which the calculation of the derivatives is then repeated. This process continues until no better poison distribution may be found. The actual formulae for varying the poison concentrations and calculating the error values are dependent upon which optimization technique is used. One of these will be examined in more detail, later, when numerical examples are presented.

All of these optimization techniques, however, will require repeated evaluations of power distributions in the same reactor, one for each poison distribution to be investigated. In order to evaluate the effects of a given poison distribution, it is first necessary to adjust its overall magnitude to obtain criticality. The flux distribution in the critical reactor must then be found, and, finally, the power distribution may be calculated. In this chapter, it will be shown how the proper choice of eigenvalue formulation and an eigenvector expansion technique can greatly aid in this part of the numerical analysis, the evaluation of the power distribution resulting from a given poison distribution.

For purposes of comparison, it is first instructive to review the approach typically used by modern computer codes

written to solve neutron diffusion problems. A good example

of such codes is CORA,[23] which was written in 1970 for

the solution of one-dimensional, few-group problems. Typi-

cally, the eigenvalue problem chosen for solution is that

of the "fission eigenfunctions," and the algorithm used is

capable of finding only the fundamental mode and its adjoint.

The equations are similar to equations (42) except that a

separate, specified control term is included. The equation

for the flux in group k at point j has the form

$$C(k,j,j+1)\phi(k,j+1)+C(k,j,j)\phi(k,j)+C(k,j,j-1)\phi(k,j-1)$$

$$+\Sigma_a(k,j)\phi(k,j)+\Sigma_R(k,j)\phi(k,j)+\varepsilon\Sigma_p(k,j)\phi(k,j)$$

$$-\sum_{n=1}^{K}\Sigma_s(n{\rightarrow}k,j)\phi_i(n,j)$$

$$=\frac{1}{K_{eff}}X(k)\sum_{n=1}^{K}\nu\Sigma_f(n,j)\phi(n,j), \qquad (73)$$

where ε is a search parameter to be explained below. The

solution of these equations is obtained by the power iteration

technique, as discussed in Chapter V. A Chebyshev polynomial

extrapolation technique is optionally available for ac-

celerating the convergence of the flux vector. Along with

the fluxes, the reactivity is found as the inverse of the

eigenvalue. A search routine is available to automatically

vary the control poison concentrations to achieve criticality.

This is accomplished by repeated solution of the fission

eigenvalue problem with all the control terms modified by

a single search parameter. This parameter, denoted as ε in equations (73), is varied between successive flux solutions according to a hyperbolic extrapolation algorithm[23] in order to reach a reactor configuration with the value of reactivity specified. Thus, the user picks the shape of the poison distribution, and the program gives the necessary magnitude.

This author would like to point out that, by selecting the poison eigenvalue formulation, instead, this same problem can be handled with only one solution of the flux equations. In order to include the option of a user specified reactivity other than unity, equations (44) for the poison eigenfluxes may be written as

$$C(k,j,j+1)\phi(k,j+1)+C(k,j,j)\phi(k,j)$$

$$+C(k,j,j-1)\phi(k,j-1)+\Sigma_a(k,j)\phi(k,j)$$

$$-\frac{1}{K_{eff}}X(k)\sum_{n=1}^{K}\nu\Sigma_f(n,j)\phi(n,j)+\Sigma_R(k,j)\phi(k,j)$$

$$-\sum_{n=1}^{K}\Sigma_s(n\rightarrow k,j)\phi(n,j) = -\varepsilon\Sigma_p(k,j)\phi(k,j). \tag{74}$$

With K_{eff} and the shape of the poison distribution, $\Sigma_p(k,j)$, specified, equations (74) may be solved by the power iteration method, with acceleration techniques if desired. The solution will be the flux distribution for the reactor configuration with the desired reactivity, and the eigenvalue,

ε, will give the necessary magnitude of the poison distribution. The need for a search for the solution of this problem may thus be eliminated by the selection of the poison eigenvalue formulation.

Finding the concentration magnitude necessary for criticality, with a given shape of the poison distribution, is only part of the overall problem of finding the optimum poison distribution to minimize power peaking. As repeatedly discussed, this total problem is nonlinear when use of a particular control material is specified. It therefore must be solved by a numerical search over various poison distributions which result in a critical reactor. Several numerical optimization techniques are more or less suitable for application to this problem. The choice is inmaterial to the topic of this paper because the methods outlined herein are suitable for use with all of them. They all require that an error value, giving the difference between the desired and actual power shapes, be found for each poison distribution investigated. The specific form of the error depends upon the method employed and how that method uses previous results to estimate the desired solution.

For each poison distribution specified by the search algorithm, the magnitude must be adjusted for criticality, the fluxes must be found, the power distribution must be computed and the appropriate error value(s) must be calculated. The utility of the poison eigenvalue formulation has already been discussed. Using this approach, the solution to a $J \cdot K$

order matrix eigenvalue equation must be found for each poison distribution investigated for a reactor represented by J space points and K energy groups. This matrix equation has the form

$$[A]\phi = -\epsilon[\Sigma_p]\phi, \tag{75}$$

where [A] contains all of the terms on the left hand side of equations (74), and $[\Sigma_p]$ contains the cross section terms for some normalized poison distribution.

The use of the poison eigenvectors to the particular reactor geometry can aid greatly in the solution of equation (75). The unknown flux vector may be written in terms of the known eigenvectors as a function of unknown expansion coefficients, a_i.

$$\phi = \sum_{i=1}^{M} a_i\phi_i, \tag{76}$$

where M is the order of expansion. If M = G·k, where G is the number of control points, then, as discussed in Chapter IV, the M poison eigenvectors will be sufficient to represent any flux distribution over the reactor that could result from an arbitrary poison distribution (with magnitude adjusted for criticality). Equation (75) may therefore be written in terms of the poison eigenvectors.

$$[A] \sum_{i=1}^{M} a_i\phi_i = -\epsilon[\Sigma_p] \sum_{i=1}^{M} a_i\phi_i \tag{77}$$

Since the matrices [A] and [Σ_p] are not functions of the index of summation, they may be included inside the sums. Substitution according to the poison eigenvalue relation, equation (45), allows the left hand side of equation (77) to be written as

$$\sum_{i=1}^{M} a_i [A] \phi_i = -\sum_{i=1}^{M} a_i \beta_i [\sigma_p] \phi_i . \tag{78}$$

The [Σ_p] matrix may be separated into the matrix, [N], giving just the normalized spatial distribution of the poison concentrations, and the matrix [σ_p], giving the cross section terms that would result from unit poison concentrations at all control points. The right hand side of equation (77) may then be written as

$$-\varepsilon [\Sigma_p] \sum_{i=1}^{M} a_i \phi_i = -\varepsilon [N] [\sigma_p] \sum_{i=1}^{M} a_i \phi_i . \tag{79}$$

After making these substitutions in equation (77), a set of M equations may be generated by multiplying both sides of the equation from the right by the transpose of one of the M adjoint vectors, ψ_m.

$$-\sum_{i=1}^{M} a_i \beta_i \psi_m^T [\sigma_p] \phi_i = -\varepsilon \sum_{i=1}^{M} a_i \psi_m^T [N] [\sigma_p] \phi_i \tag{80}$$

The orthogonality relations for the poison eigenvectors and their adjoints are found from equation (37) by substituting [σ_p] for [B].

$$\psi_m^T [\sigma_p] \phi_i = \delta(i,j) \tag{81}$$

Using these relations, equations (80) may be simplified to

$$a_m \beta_m = \varepsilon \sum_{i=1}^{M} a_i F(m,i,[N]), \qquad \text{for } m = 1 \text{ to } M, \tag{82}$$

where

$$F(m,i,[N]) = \psi_m^T [N] [\sigma_p] \phi_i. \tag{83}$$

If $M = G \cdot K$, then solution of the M "component equations" (82) is completely analogous to the solution of the unexpanded equations (74). Equations (82) may be written together as a single matrix equation.

$$[\beta]\overline{a} = \varepsilon [F]\overline{a}, \tag{84}$$

where $[\beta]$ is a diagonal matrix, with all of the β_m on the diagonal, and $[F]$ is a full M by M matrix with elements given by equation (83).

The number of control points, G, may be substantially less than the total number of points, J. Thus, the substitution of a $G \cdot K$ order matrix equation for a $J \cdot K$ order equation appears to offer a substantial savings in computational work for evaluating the power distribution produced by each poison distribution considered. In addition, $[\beta]$ is diagonal and is more easily inverted than is the $[A]$ matrix of the unexpanded problem. However, the eigenvalue nature of equation (84) presents a problem. Generally, there are as many

as M solutions to the equation, just as there are to the
eigenvalue equation (45) used to generate the M poison eigen-
functions. As pointed out in Chapter V, a simple power
iteration solution of the equation

$$\overline{a}/\varepsilon = [\beta]^{-1}[F]\overline{a} \tag{85}$$

will converge to   whatever mode has the ε with absolute
value closest to zero. In general, there is no reason to
expect the fundamental mode to meet this criterion. The
fundamental mode, with the smallest neutron losses due to
leakage, requires the greatest amount of poison to attain
criticality, and therefore the most positive value of ε. Ob-
viously, this problem also occurs for the unexpanded poison
eigenvalue formulation, as given by equation (75). There
are two approaches to insuring that power iterations converge
to the fundamental mode of equation (84). Both have their
advantages and disadvantages.

The more obvious approach involves the use of the
Wielandt transformation. An initial guess is made for the
proper value of ε to produce the fundamental mode for the
first poison distribution to be considered. Then, as the
computer program proceeds in its search from one poison
distribution to the next, the eigenvalue found for the last
solution is used as the guess for the next. If the program
does not drastically change the poison distribution from one
time to the next, the fundamental modes to the two different
reactor configurations should be similar enough, with eigenvalues

that are close enough, that this method may be expected to work quite well. In practice, no difficulty has been experienced by this author from the interference of other modes. The starting guess for ε in the first poison distribution case could be made in several ways. A very large value of ε could be chosen to insure that it is greater than all the actual roots. However, the safer this guess is made, the worse the dominance ratio becomes and the more iterations are required to reach convergence. Another method is to start the poison search with a uniform distribution. In this case, $[\Sigma_p] = [\sigma_p]$, and the unexpanded poison search equation (75) becomes identical to the poison eigenfunction equation (45). Thus, the desired value of ε is identical to β_1, the eigenvalue of the first poison eigenfunction. This method for predicting ε, while accurate, does not allow any prior knowledge or intuition of the optimum to be used in specifying the starting distribution for the poison search. A third method, which the author has found satisfactory, is to use first order perturbation theory to predict the value of ε for the fundamental mode associated with the desired starting poison distribution. If the flux for this mode is assumed to be approximately equal to the flux associated with a uniform poison distribution, then the first fission eigenflux may be substituted for the flux in equation (75).

$$[A]\phi_1 = -\varepsilon[N][\sigma_p]\phi_1 \tag{86}$$

Using eigenvalue relationship for ϕ_1, this becomes

$$-\beta_1 \phi_1 = -\varepsilon [N][\sigma_p]\phi_1. \tag{87}$$

Multiplying both sides by the transpose of the adjoint flux, ψ_1^T, gives a scalar equation which may be solved for ε to yield

$$\varepsilon = \psi_1^T [N][\sigma_p]\phi_1/\beta_1 = F(1,1,[N])/\beta_1. \tag{88}$$

Since $F(1,1,[N])$ has already been found for use in the expansion equations (84), this method provides a simple means for generating a reasonably accurate guess at the fundamental mode ε. Although there is no guarantee that a guess generated by equation (88) will be closest to the maximum positive value of ε, this author has experienced no difficulty in practice.

The disadvantage of the Wielandt transformation approach is that the transformed equation has the form

$$\{[\beta]-\varepsilon_g[F]\}\bar{a} = \delta[F]\bar{a}. \tag{89}$$

Thus, a full M order matrix must be inverted, instead of the simple diagonal matrix $[\beta]$, for the power iteration with the transformed operator $\{[\beta]-\varepsilon_g[F]\}^{-1}[F]$. In comparing the unexpanded and expanded formulisms for complexity of the power iteration process, note must be made of the special nature of the [A] matrix in the unexpanded equation (75). The exact form of [A] will depend upon how many spatial

dimensions and energy groups have been used and upon the details of scattering among the energy groups. In particular, if only one spatial dimension is considered, the multi-diagonal nature of [A] (and therefore of $[A] - \varepsilon_g[\Sigma_p]$) lends itself to the use of "inner iteration schemes" that take the place of the matrix inversion.[23,24] The decision of which is more complicated, the use of such an inner iteration scheme on a $J \cdot K$ matrix, or the generalized inversion of a $G \cdot K (=M)$ matrix, must depend upon the relative magnitudes of G and J and upon the exact nature of the [A] matrix for each particular case. If more than one spatial dimension is considered, solution of the unexpanded equation becomes much more difficult. The "inner iterations" may be done in each direction independently, and the directions alternated until convergence of the entire set of inner iterations is obtained.[16] In such cases, however, inversion of the $G \cdot K$ matrix would certainly seem preferable unless G is very nearly as large as J.

The second method for insuring that the power iteration method converges to the proper value of ε is designed to preserve the [β] matrix in a simply invertable form. This may be accomplished by observing the following phenomenon. Any eigenvector, ϕ_i, that is a solution to equation (45) will also be a solution to the equation

$$\{[A] + \beta_s[\sigma_p]\}\phi_i = -(\beta_i - \beta_s)[\sigma_p]\phi_i = -\beta_i'[\sigma_p]\phi_i. \qquad (90)$$

That is, reducing all of the β_i by the amount β_s is equivalent

to assuming that the concentration of control poison is already β_s at all control points. If β_s is made large enough that the fundamental poison eigenvalue, β_1', is negative, then physically, the reactor is shut down. The problem of determining the optimum poison distribution then becomes a matter of removing poison from what is already there.

The distributions investigated by the search program will be the complementary distributions necessary to make the actual distributions equal to β_s at every control point. That is, for normalized actual distribution [N], and complementary distribution [N'], the relationship between the magnitudes necessary for criticality of the fundamental mode is

$$\epsilon[N][\sigma_p] = \epsilon'[N'][\sigma_p]+\beta_s[\sigma_p]. \tag{91}$$

The flux matrix equation associated with the evaluation of such complementary distributions, [N'], is

$$\{[A]-\beta_s[\sigma_p]\}\phi = -\epsilon'[N'][\sigma_p]\phi. \tag{92}$$

Using the same set of poison eigenvectors and adjoints and the new eigenvalue relations (90), equation (92) may be expanded by a method analogous to that used in obtaining the expanded equation (84).

$$[\beta']\overline{a} = \epsilon'[F']\overline{a} \tag{93}$$

where the elements of [F'] are given by analogy to equation

(83) as

$$F'(m,i,[N']) = \psi_m^T[N'][\sigma_p]\phi_i. \tag{94}$$

Thus, the value of ε' which corresponds to the fundamental mode may be varied by varying β_s.

As long as poison is only removed at each point, there is no ambiguity. However, if the base level of control poison, β_s, is such that some near optimum trial distribution has values greater at some points, and less at others, then the complementary distribution to be removed will be positive at some points and negative at others. Such a distribution may be interpreted two ways. With a negative ε', the trial distribution is to be subtracted from the uniform distribution, but with a positive ε', it is to be added. Since the trial distribution has both positive and negative elements, for either case, poison will be added at some points in the reactor and removed at others. If the complementary distribution chosen for investigation is such that both positive and negative roots for ε' exist, then it is impossible to specify beforehand that the distribution is to be added or subtracted. The power iteration solution method will always converge to whichever value of ε' is closest to zero.

Computer search programs usually cannot allow for more than one interpretation of the set of search parameters (in this case, [N']) chosen for evaluation of the error function. Therefore, in order to insure that such an ambiguous situation does not arise, it is prudent to chose the value of β_s

so that, for all total poison distributions that may reasonably be considered, the complementary poison distribution will be everywhere negative. The search program may then be expected to restrict itself to trial distributions that everywhere have the same sign, thus insuring that all the values of ε' are of the opposite sign. (The actual sign of [N'] itself is of no significance because only the product, $\varepsilon'[N]$, must be negative to designate removal of poison.) The fact that such a selection of β_s will result in all possible root values of ε' being of the same sign is physically obvious. Since the reactor is shutdown by the uniform poison distribution, β_s, and since poison may only be removed everywhere or added everywhere by the distribution [N'], obviously, β_1', the fundamental mode root must be of the proper sign to remove poison if that mode is to be critical. Other roots must correspond to more buckled flux shapes, and require even more poison to be removed for criticality of their modes. Thus, the other values of ε' must be greater in absolute value and of the same sign as ε_1'. This insures that the power iteration solution of equation (91) will always be the fundamental mode and that the distribution [N'] will always be considered as subtractive.

Unfortunately, equation (91) does not aid in the selection of a value for β_s that will insure behavior for ε' as described above. Since the "worst case" trial distribution, $\varepsilon[N]$, will not be known before the search program is

run, some other means must be used to select β_s. Certainly, β_s must be greater than β_1, or the reactor will not be initially shut down. If β_s is chosen extremely large, then it is also certain that poison would have to be removed everywhere to reach the critical state with the optimum power distribution. However, such large guesses for β_s also serve to increase the absolute magnitudes of all the β_i'. This tends to decrease the dominance ratio for β_1', causing more iterations to be required for convergence of the flux for each $[N']$ investigated. For the general case, this may be hard to grasp, but for the special case $[N'] = [I]$, it is easily illustrated. In that case, $\varepsilon_i = \beta_i$, and the subtractive distribution $[N] = [I]$. This allows equation (91) to be simplified to

$$\beta_i = \varepsilon_i' + \beta_s. \tag{95}$$

Thus, the dominance ratio for the ε_i', as a function of β_s, is $\left| (\beta_2 - \beta_s)/(\beta_1 - \beta_s) \right|$. Since $\beta_2 < \beta_1 < \beta_s$, this dominance ratio for ε_1 must decrease as β_s increases (i.e., as the reactor is further shut down).

In order to use this method for insuring that the proper root is obtained for ε in the solution of the expanded flux equations, it is therefore necessary to make an educated guess at the best value of β_s. If the value chosen is too small, the error may be detected when a value of ε_1' with an improper sign is computed, or when $[N']$ contains negative elements. In such an event, it is necessary that the value

of β_s be increased. The progress toward the optimum need not be lost, however. From equation (91), it is possible to find the new complementary poison distribution associated with the new value of β_s, that corresponds to the best distribution found so far using the old value of β_s. In general, [N] and $[\sigma_p]$ will be diagonal with some zero diagonal elements in positions corresponding to noncontrol points. If these zero rows and columns are eliminated, however, then the resulting matrices will possess inverses. With the understanding that this has been done, the solution for the total poison distribution associated with the best complementary distribution, $\epsilon'(1)[N'(1)]$, found so far using $\beta_s(1)$, is

$$\epsilon[N] = \epsilon'(1)[N'(1)] + \beta_s(1)[I], \tag{96}$$

where the fact that the reduced $[\sigma_p]^{-1}$ is diagonal, and thus commutative under matrix multiplication, has been used to simplify the relation. The new complementary distribution, $\epsilon'(])[N'(2)]$, must represent this same total distribution. Equating the two expressions for $\epsilon[N]$ and solving yields

$$\epsilon'(2)[N'(2)] = \epsilon'(1)[N'(1)] + \{\beta_s(1) - \beta_s(2)\}[I]. \tag{97}$$

This new distribution, normalized to $[N'(2)]$, may be used as the new starting point when the optimization search is begun again using $\beta_s(2)$. This process of retransforming the β set and restarting the optimization search could be done manually or automatically by the search program.

If an unnecessarily large value of β_s is picked, however,

there will be no way of knowing this for certain until the optimization search is completed. This is too late to improve convergence speed by decreasing β_s to increase the dominance ratio. Perhaps a routine could be programmed that would automatically decrease β_s if previously found values of ε_i' have been consistently large. In lieu of such a scheme, or perhaps in addition, it would probably be worthwhile to include a flux convergence acceleration scheme to help make up for the expected poor dominance ratios. An example of this type of acceleration scheme is the Chebyshev polynomial extrapolation method utilized by the CORA program.

Of the two methods described above for the solution to the poison distribution evaluation problem, the first, using the Wielandt transformation, is the easier to program and understand. Because of the necessity of inverting the $[F]$ matrix, however, its utility may be limited. The second method, involving only the linear modification of the set of poison eigenvalues, seems to offer much greater computational power. This method requires complex programming to be made economical, however.

CHAPTER VII

NUMERICAL CALCULATIONS AND THE PROSPECT
FOR PARTIAL EXPANSIONS

In order to illustrate the use of some of the tech-
niques described in previous chapters, their application
to several sample problems will now be presented. This
will serve two purposes. First, some of the principles
previously discussed only in the abstract sense will be
demonstrated by concrete example. These calculations are
then extended to explore the possibility of using less than
the complete set of "component equations" (82) in order
to approximate the solution with less work.

The principles of eigenvector generating programs
were discussed in Chapter V and Appendix I, and a sample
program is listed in Appendix II. The optimization search,
as discussed in Chapter VI, is carried out by the use of a
specially modified version of the GAUSS computer code written
by Kylstra* at the University of Florida. This program is
based on the least squared error technique of Gauss,[25] the
principles of which are described in Appendix III. The
eigenvector expansion technique, which is used for synthesi-
zing the flux and power distributions for each trial poison

_____

*C.D. Kylstra, personal communication.

distribution generated by GAUSS, is incorporated in the subroutine UFUNT. Listings of this subroutine and the auxiliary input and output subroutines are given in Appendix IV. The value of the error function associated with each trial poison concentration is calculated by the algorithm

$$\text{ERROR} = \sum_{j} \{1-P(j)\}^2 V(j), \tag{98}$$

where $P(j)$ is the (normalized) power density at point j and $V(j)$ is the volume over which this power density applies. The sum runs over all points j in fueled regions. It is this sum that is minimized by the GAUSS program. In cases where the minimum error is not zero, it may be seen from equation (98) that the solution given is the optimum in the least squared errors sense. Other optimization techniques, requiring different formulations of the error function, could be used if desired. The Gaussian technique was chosen because of its reputation for providing solutions in complicated cases where other methods may fail, and due to its availability to this author in a generalized computer code.

Because of its relatively simple programming requirements, the Weilandt transformation approach has been used to ensure that the proper ε root is found in solving the expanded equation (84). Although, as discussed in Chapter VI, this method may be more limited in its utility than is the β_s transformation method, it is perfectly adequate for

the purpose of demonstrating the eigenvalue expansion tech-
nique. In order to gain a slight simplification in the neces-
sary matrix algebra, the equation solved by the UFUNT sub-
routine has been rewritten from equation (84).

$$[F]\overline{a} = (1/\epsilon)[\beta]\overline{a} \tag{99}$$

Using the Wielandt transformation with a guess at the inverse
of the desired eigenvalue, $(1/\epsilon)$, this equation may be
written as

$$\{[F]-(1/\epsilon)_G[\beta]\}\overline{a} = \delta[\beta]\overline{a}, \tag{100}$$

where $\delta = (1/\epsilon)-(1/\epsilon)_G$. The matrix operator used for itera-
tion purposes is thus $\{[F]-(1/\epsilon)_G[\beta]\}^{-1}[\beta]$.

In order to provide a check of the optimization program,
a simple case, amenable to two solution techniques, was
chosen as the first example. A bare, homogeneous infinite
half-slab reactor was represented by a five-point, two-energy
group model. The materials parameters for Example 1 are
given in Table 1. In this model, there are ten flux values
and, with each point designated a control point, there are
ten poison eigenvectors. Because the energy and space
variables are separable in diffusion theory models of bare,
homogeneous reactors, these vectors may be associated as
pairs, with identical shapes but different (spatially con-
stant) fast/thermal flux ratios. The first two pairs of
modes are illustrated in Figure 2 and their adjoints are
shown in Figure 3. The ten eigenvalues are listed in Table 2.

TABLE 1

Reactor Model for Example 1

Core Configuration: bare, homogeneous, infinite slab,
    80 cm total thickness.

Mathematical Model: diffusion theory, 2 energy groups,
    5 space points representing half-slab, mirror condi-
    tion at centerplane (point 1), zero flux condition at
    extrapolated boundary (point 6).

Materials Parameters (in standard metric units):

Group (k)	$D(k)$	$\Sigma_a(k)$	$\Sigma_R(k)$	$\nu\Sigma_f(k)$	$\chi(k)$
1	1.46	.0122	.0175	.0126	1
2	0.37	.1841	--	.2666	0

    lattice spacing = 8 cm

Control Poison Cross Sections:
    $\sigma_p(1) = 0.1$ cm^{-1} per gram/cc
    $\sigma_p(2) = 0.2$ cm^{-1} per gram/cc
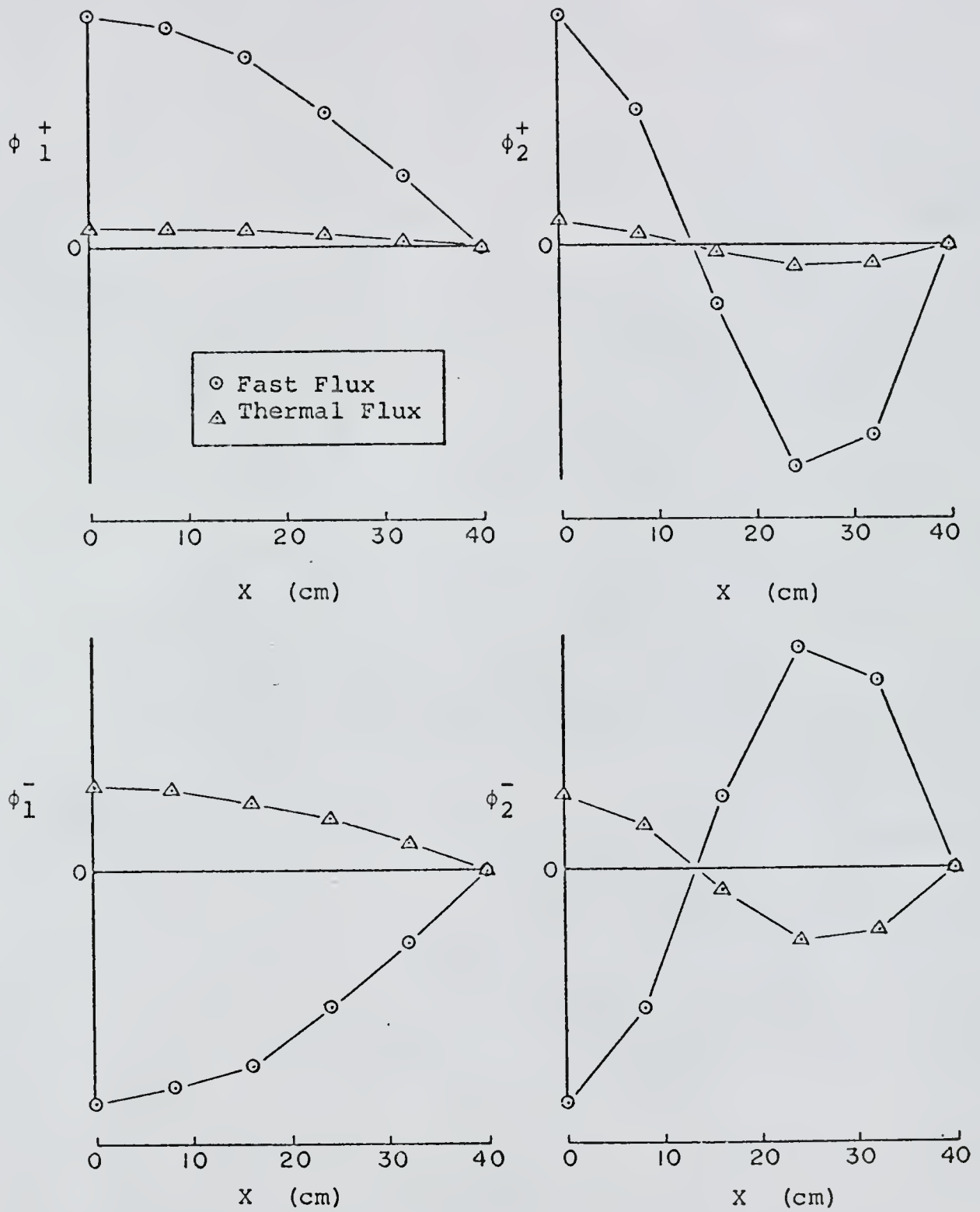
Figure 2   Poison Eigenvectors ϕ_1^+, ϕ_1^-, ϕ_2^+ and ϕ_2^- for Example 1
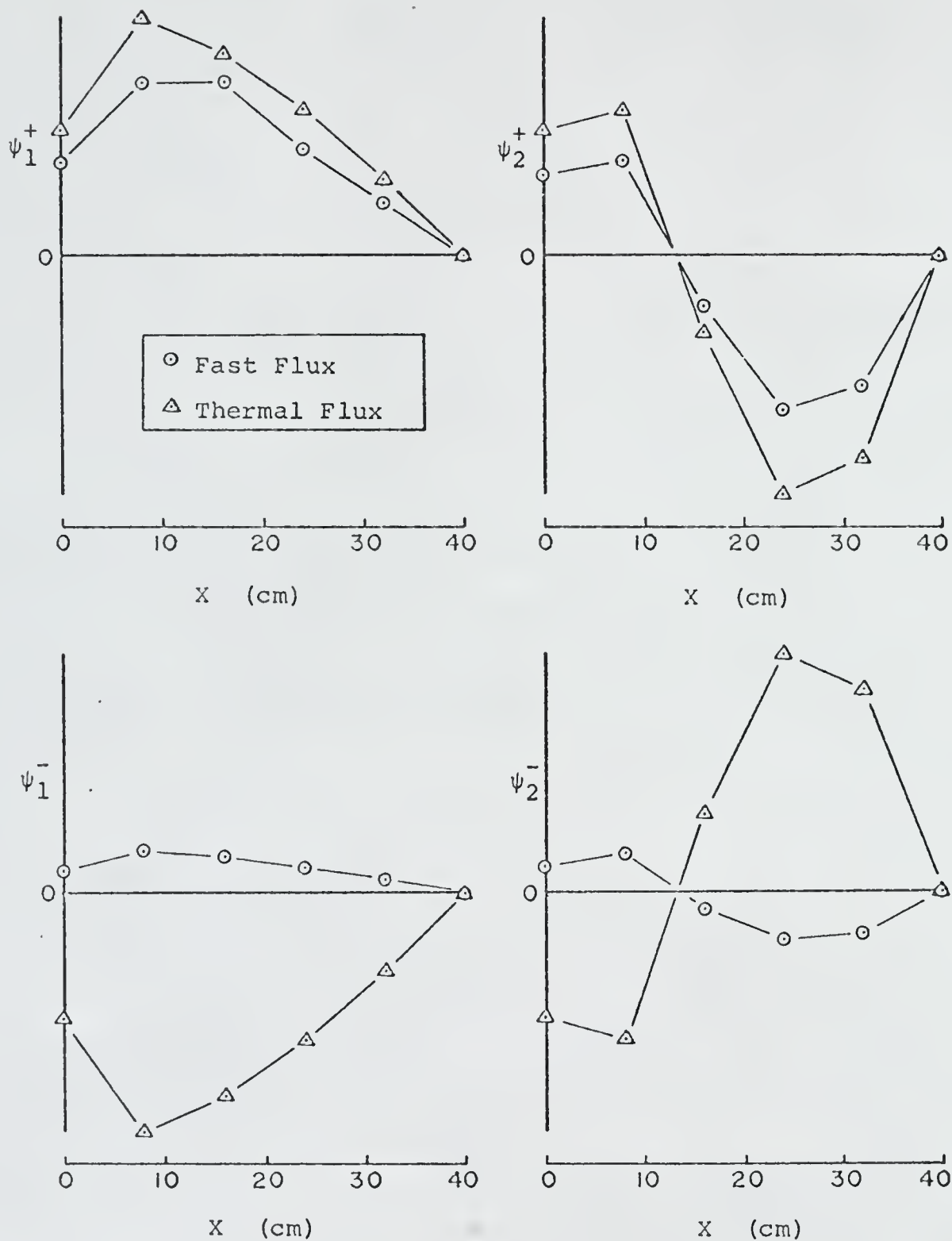
Figure 3   Adjoint Vectors ψ_1^+, ψ_1^-, ψ_2^+ and ψ_2^- for Example 1

TABLE 2

The Eigenvalues Associated with the Poison
Eigenvectors for Example 1

Mode Index	Eigenvalue	Mode Index	Eigenvalue
1+	0.047070	1-	-1.163727
2+	-0.086924	2-	-1.216454
3+	-0.288781	3-	-1.316718
4+	-0.467260	4-	-1.440360
5+	-0.563996	5-	-1.530345

The behavior discussed in Chapter V for the poison eigen-value series may easily be seen in this example. Because the fundamental mode in the unpoisoned reactor is super-critical, β_1^+ is positive. All the other β^+ grow more nega-tive with increasing index. The β^- series are all negative, and also become more negative with increasing index. It is also apparent that, if enough points were added to the model, the two series may overlap. That is, for some index i great enough, there would be $\beta_i^+ < \beta_1^-$. This is one of the phenomena which complicates any automatic process for finding the whole set of eigenvalues with no human intervention.

The pointwise optimum poison concentrations for Example 1 were found by the modified GAUSS program, using the tech-niques outlined in Chapter VI. The results are displayed in Table 3.

In order to provide a check on the functioning of the optimization coding, this problem was also solved by the following alternative method. A power distribution vector was associated with each of these poison eigenvectors, and is designated by P_i^+ for the ϕ_i^+ mode. These power vectors have only spatial dependence, and thus only five elements each. The elements of the power vectors, $P_i^+(j)$, are related to the elements of the flux vectors, $\phi_i^+(k,j)$, by the equa-tion

$$P_i^+(j) = A\{\Sigma_f(1)\phi_i^+(1,j) + \Sigma_f(2)\phi_i^+(2,j)\}$$

$$= A\{\Sigma_f(1)S_i^+ - \Sigma_f(2)\}\phi_i^+(2,j). \tag{101}$$

TABLE 3

Optimization Results for Example 1

Results of General Poison Distribution Search Method:

Point No.	Fast Flux	Thermal Flux	Power Density	Poison Concentration
1	2.75802E 01	2.44744E 00	1.0	6.55390E-02
2	2.75777E 01	2.44756E 00	1.0	6.54151E-02
3	2.75518E 01	2.44878E 00	1.0	6.41074E-02
4	2.73029E 01	2.46055E 00	1.0	5.16011E-02
5	2.49391E 01	2.57227E 00	1.0	-1.02413E-01

Results of Power Vector Expansion Coefficient Search Method:
  Real Solution:

Point No.	Fast Flux	Thermal Flux	Power Density	Poison Concentration
1	2.75802E 01	2.47444E 00	1.0	6.55372E-02
2	2.75778E 01	2.47555E 00	1.0	6.54153E-02
3	2.75518E 01	2.48788E 00	1.0	6.41076E-02
4	2.73029E 01	2.40555E 00	1.0	5.16005E-02
5	2.49391E 01	2.52266E 00	1.0	-1.02412E-02

Mode Index	Expansion Coefficient	Mode Index	Expansion Coefficient
1+	2.72001E 01	1-	3.82306E-02
2+	-8.81539E 01	2-	1.49576E-03
3+	4.88696E 01	3-	-1.66447E-01
4+	-2.71797E 01	4-	2.21372E-01
5+	8.85565E 01	5-	-9.48818E-02

Second Mathematical Solution (Not Physically Possible):

Point No.	Fast Flux	Thermal Flux	Power Density	Poison Concentration
1	-1.19011E 01	4.31338E 00	1.0	-1.16233E 00
2	-1.12463E 01	4.28244E 00	1.0	-1.14970E 00
3	-1.23898E 01	4.33649E 00	1.0	-1.17165E 00
4	-9.86252E 00	4.21704E 00	1.0	-1.12248E 00
5	-1.54869E 01	4.48286E 00	1.0	-1.25350E 00

Mode Index	Expansion Coefficient	Mode Index	Expansion Coefficient
1+	4.48286E 02	1-	-1.54869E 03
2+	4.21704E 02	2-	-9.86252E 02
3+	4.33649E 02	3-	-1.22898E 03
4+	4.28244E 02	4-	-1.13463E 03
5+	4.31338E 02	5-	-1.19011E 03

A similar series of power vectors, P_i^-, is defined for the ϕ_i^- series of modes. Since ϕ_i^+ and ϕ_i^- have the same spatial shape, and differ only by their fast/thermal flux ratios, S_i^+ and S_i^-, their associated power vectors, P_i^+ and P_i^-, differ only in magnitude. That is, $P_i^+ = C_i P_i^-$.

The "desired power shape" is everywhere constant, and, when normalized to a unit power density, is given for this example by the vector $P^* = (1,1,1,1,1)$. If the flux in the reactor, ϕ^*, that results in this power shape is expanded as a sum of the poison eigenvectors,

$$\phi^* = \sum_{i=1}^{5} a_i^+ \phi_i^+ + a_i^- \phi_i^-, \tag{102}$$

then the desired power vector, P^*, is given by

$$P^* = \sum_{i=1}^{5} a_i^+ P_i^+ + a_i^- P_i^- = \sum_{i=1}^{5} (a_i^+ + a_i^-/c_i) P_i^+. \tag{103}$$

As may be seen from equation (103) the set of five P_i^+ vectors is alone sufficient to expand the desired power vector. Such an expansion has the form

$$P^* = \sum_{i=1}^{5} b_i P_i^+. \tag{104}$$

The vector equation (104) may be solved by simple matrix inversion to obtain the b_i. By equating coefficients for P_i^+ in equations (103) and (104), the following relations are derived.

$$(a_i^+ + a_i^-/c_i) = b_i \qquad (105)$$

If the five new variables, d_i, are introduced such that

$$a_i^+ = b_i/d_i, \qquad (106)$$

then, from equation (105),

$$a_i^- = c_i b_i (d_i - 1)/d_i. \qquad (107)$$

Thus, the choice of the five variables, d_i, allows the computation of a set of ten flux expansion coefficients, a_i, which allow the construction of a flux satisfying the flat power condition. With these coefficients and the poison eigenvectors, the poison concentrations necessary for criticality may be found by solving equation (77). After making the substitutions given by equations (78) and (79), the solution for each diagonal element of the [N] matrix becomes

$$\varepsilon N(k,j) = \sum_{i=1}^{10} a_i \beta_i \phi_i(k,j) / \sum_{i=1}^{10} a_i \phi_i(k,j). \qquad (108)$$

However, the concentrations for both groups must be the same at any one point, so there are five constraint equations.

$$N(1,j) = N(2,j) \qquad (109)$$

Unfortunately, equations (108) are nonlinear in the d_i when these variables are substituted for the a_i. The problem was solved by an optimization search on the variables d_i, with the optimum point defined by the satisfaction of the constraint equations (109).

This search was carried out by an unmodified version of the GAUSS program, and the results are given in Table 3. As may be seen, both methods result in the same solution, indicating that the general program does indeed find the optimum poison distribution. The poison eigenvectors for this example satisfy their orthogonality relations to the degree that the products $\psi_m^T[\sigma_p]\phi_i$, for $m \neq i$, have values on the order of 10^{-6} or less. With this fact in mind, it may be seen from the expansion coefficients for the optimum flux, given in Table 3, that all of the modes give significant contributions to the expansion. The modes with negative fast/thermal flux ratios give only small contributions in this case. More will be said about the significance of this series of modes later in the chapter, when partial expansions are considered.

Before leaving Example 1, it is useful to point out that another solution exists to equations (108) and (109). This mathematical solution has negative fast flux components, and may therefore be excluded from consideration on physical grounds. However, it serves to illustrate the fact that the nonlinearity of the mathematics does allow for more than one solution, as discussed in Chapter III. Because a two-group formulation was used in this example, two mathematical solutions exist. This nonreal solution is also included in Table 3.

Example 2 was run in order to provide an independent check of the CODE2 eigenvector generator and the GAUSS-UFUNT

eigenvector expanded poison optimization routine. This example employs a two-group, ten-point model for a bare, homogeneous, infinte cylindrical reactor. The reactor configuration from Example 2 is given in Table 4.

The fundamental poison eigenvalue and vector were checked against a calculation performed by the CORA multi-group diffusion theory program in which a uniform poison concentration, β_1, had been added to the materials. The eigenvector, the CORA generated fluxes, and the analytic solution are all plotted in Figure 4. As may be seen, the comparison is excellent. Because the CODE2 and CORA values occur at the same radial locations, and the two sets of values are identical to five significant digits, only one set of points is shown. In addition, the reactivity of the flux mode calculated by CORA was within 10^{-5} of unity, the value assumed in the generation of the poison eigenvectors. For the unpoisoned core, CORA and the analytic solution give a reactivity of 1.0603.

Because the CORA program is not capable of handling pointwise variations in the poison cross section, the optimization was performed for three regions of poison. The poison concentrations at the control points were constrained to vary as three groups in the manner described in Table 4. Because CORA requires that materials interfaces occur on mesh points, it was necessary to locate the boundaries of the constant poison regions at the mesh points used for the eigenvector calculations, in order to preserve identical

TABLE 4

Reactor Model for Example 2

Core Configuration: bare, homogeneous, infinite cylinder, 150 cm radius

Mathematical Model: diffusion theory, 2 energy groups, 10 space points along radius, zero current condition at centerline (point 1), zero flux condition at extrapolated boundary (point 11)

Materials Parameters (in standard metric units):

Group (k)	$D(k)$	$\Sigma_a(k)$	$\Sigma_R(k)$	$\nu\Sigma_f(k)$	$X(k)$
1	1.461860	0.0103369	0.0169434	0.005872	1
2	0.366341	0.0979940	–	0.135760	0

lattice spacing = 15 cm

Control Poison Cross Sections:
$\sigma_p(1) = 2.86715$ cm^{-1} per gm of B^{10}/cc
$\sigma_p(2) = 130.693$ cm^{-1} per gm of B^{10}/cc

To achieve regionwise poison variation:
points 1 through 3 are constrained to the same value,
points 5 and 6 are constrained to the same value,
points 8 through 10 are constrained to the same value,
point 4 is constrained to the volume weighted average
        of the values in the inner and middle regions,
point 7 is constrained to the volume weighted average
        of the values in the middle and outer regions.
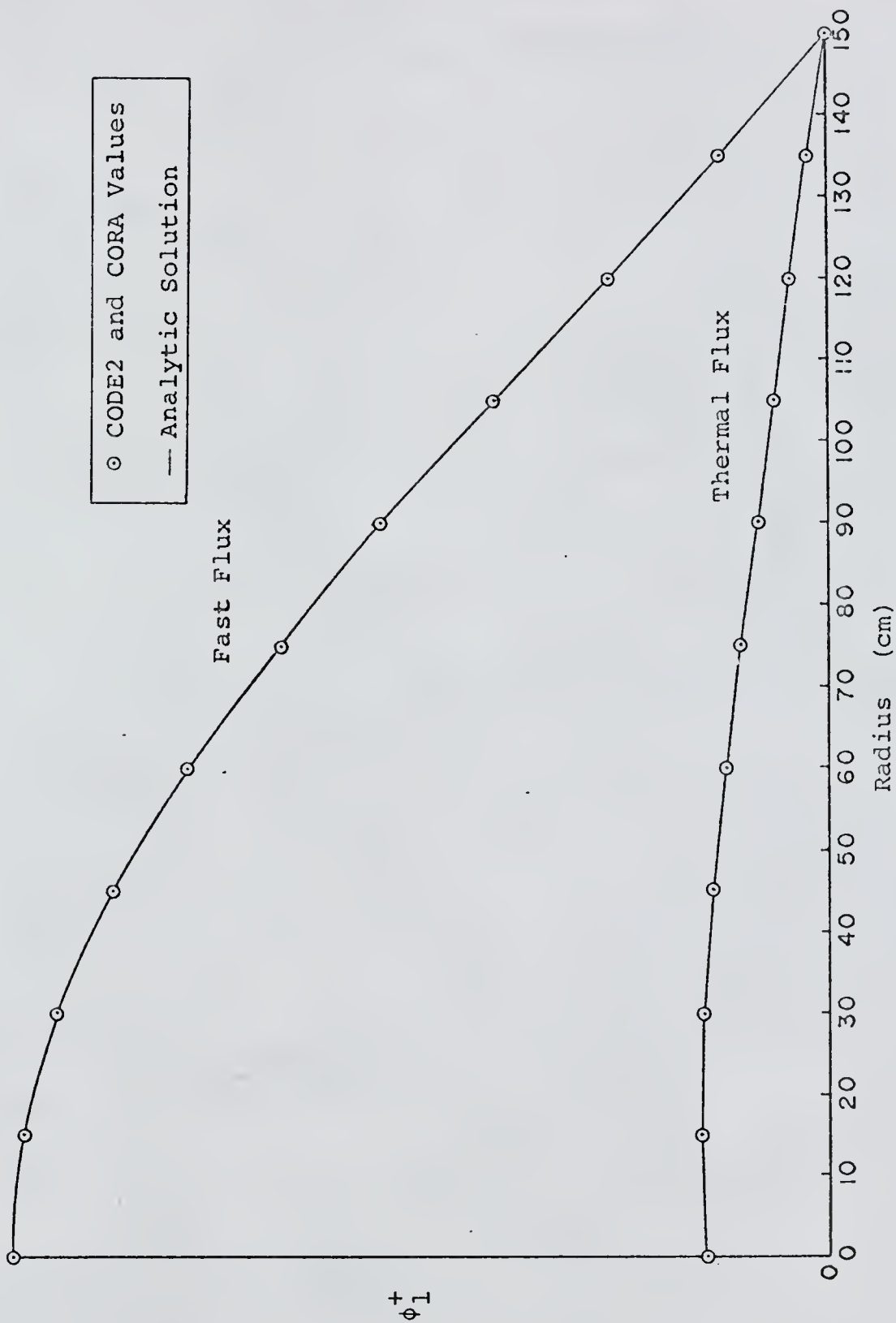
Figure 4 Comparison of the Poison Eigenvector, ϕ_1^+, for Example 2 as Calculated by CODE2, by CORA and Analytically

mathematical models between the two calculations. In

order to do this, the concentrations at the interface points

were taken as the average of the concentrations in the re-

gions to either side, weighted by the fraction of the reac-

tor volume that is in each region and associated with that

mesh point. The results of the optimization are given in

Table 5. A comparison calculation was then made for this

optimum configuration using the CORA program. The results

show excellent agreement. The flux values are identical to

at least three, usually four significant digits. These

values are plotted in Figure 5. The CORA calculated reac-

tivity for this materials configuration is 1.0000114, which

is also in excellent agreement with the value of unity

assumed in using the eigenvector expansion technique.

It should be pointed out that regionwise poison varia-

tion may lead to more than one "optimum realizable power

distribution". This is due to the fact that the error

value, given by equation (98), is then a function of more

points than there are available control parameters. The

problem is thus underspecified, and the minimum value of

the error will not, in general, be zero. If the poison

concentrations available for manipulation by GAUSS are

considered as the ordinates of a "concentration hyperspace",

then the error function may be considered a hypersurface

within that space. As shown already by the two solutions

to Example 1, there are more than one minimum on this error

hypersurface for a completely specified problem. (All but

TABLE 5

Three-Region Poison Optimization Results for Example 2

Poison Region	Point No.	Poison Concentration	Power Density	Volume Weight
I	1	4.66710E-05	1.20655	2.77008E-03
I	2	4.66710E-05	1.18203	2.21607E-02
I	3	4.66710E-05	1.10942	4.43213E-02
I	4-	4.66710E-05	0.97136	3.04708E-02
II	4+	9.68959E-05	0.97136	3.60111E-02
II	5	9.68959E-05	0.90423	8.86427E-02
II	6	9.68959E-05	0.97290	5.30931E-02
II	7-	9.68959E-05	1.19125	5.77099E-02
III	7+	7.47632E-06	1.19125	1.32964E-01
III	8	7.47632E-06	1.32066	1.55126E-01
III	9	7.47632E-06	1.06908	1.77285E-01
III	10	7.47632E-06	0.58144	1.99446E-01
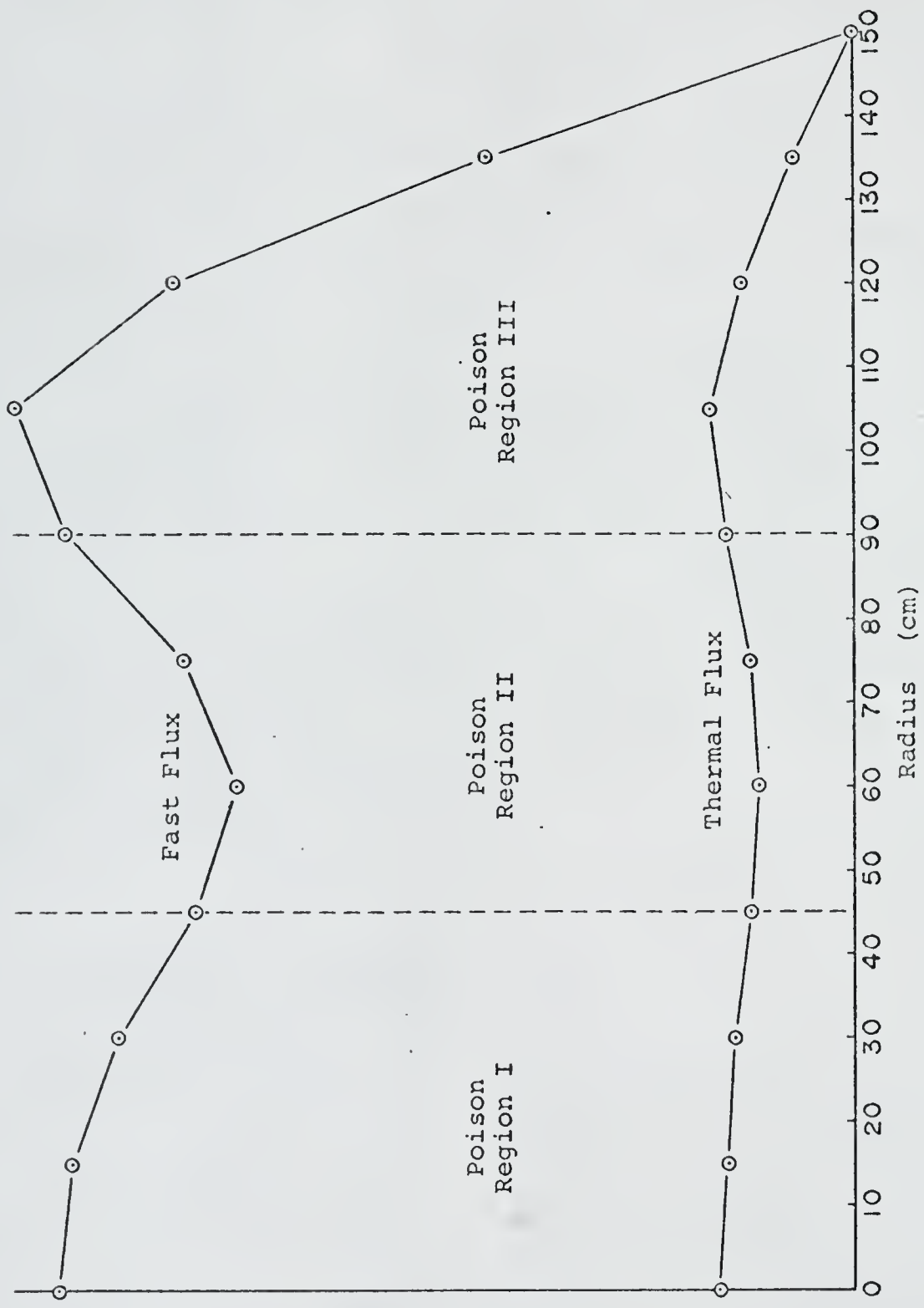
Figure 5   Flux Distribution for the Optimum Three-Region Poison Distribution in Example 2

one of these may be eliminated by physical considerations,
however.) If the problem becomes underspecified, the hyper-
space has fewer dimensions and the error surface may develope
extra local minima. This is a phenomenon that is caused
by the physically underspecified nature of the problem,
rather than by the techniques used for solving that problem.
Underspecified problems may be created by ganging control
points for regionwise variation or by the specification of
some fueled points that are not control points. In any
event, no matter how these problems are treated, it must be
remembered that an optimization search program may find a
local minimum that is not the lowest minimum. Indeed,
there may be more than one lowest minimum, although this is
unlikely.

Now that it has been established that the eigenvector
expansion technique may be expected to give a reasonable
representation of the unexpanded problem, the possibility
of using expansions with less than the complete number of
vectors may be investigated. Because Examples 1 and 2
both involve cases where the energy and space variables
are separable, partial expansions of these problems are
special cases. If the vectors are eliminated in the pairs,
ϕ_i^+ and ϕ_i^-, some of the ability to represent spatial shape
has been lost. However, any shape that may be generated by
the remaining vectors may have its (spatially dependent)
spectrum multiplied by any arbitrary magnitude. The spectrum

of each shape component is still arbitrarily adjustable

between the two vectors with that shape. For a reactor

with materials interfaces, the energy and space variables

are no longer separable, and the vectors ϕ_i^+ and ϕ_i^- will

no longer share the same shape. In such a case, elimination

of some of the vectors may also impare the representation

of the energy spectrum of the flux shapes which may still

be generated.

In order to investigate partial expansions for general

multigroup problems, Example 3 was contrived. The model is

a two-group, nineteen-point representation of an infinite

cylindrical reactor with two fueled regions and a water

reflector. The details of this reactor model are given in

Table 6. The fifteen control points result in thirty poi-

son eigenvectors. The ϕ_1^+ and ϕ_1^- modes are displayed in

Figure 6 to illustrate their dissimilarity in shape. In

addition, it may be seen that the fast/thermal flux ratio

is now a function of position in the reactor for each mode.

The adjoint vectors for these modes are shown in Figure 7.

Partial expansions can be expected to be of the most

utility when the poison is to be varied by region. In such

cases, if ever, a few modes might be expected to adequately

represent the somewhat rippled flux shape that results. How-

ever, because regionwise variation of poison involves step

changes in poison concentration, all of the eigenvectors

may still be needed to represent such situations with maximum

accuracy. That is, although contributions from the higher

TABLE 6

Reactor Model for Example 3

Core Configuration: infinite cylinder, two homogeneous
    fueled regions from radii 0 to 50 cm and 50 to 140 cm,
    pure water reflector from radius 140 to 190 cm
Mathematical Model: diffusion theory, 2 energy groups,
    19 space points, 15 control points, zero current
    condition at centerline (point 1), zero flux condition
    at extrapolated boundary (point 20), continuous neutron
    current conditions at internal interfaces (points
    6 and 15)

Materials Parameters (in standard metric units):

Region I: Westinghouse PWR fuel assemblies, 3% enrichment,
    18,000 MWD/T

Group(k)	$D(k)$	$\Sigma_a(k)$	$\Sigma_R(k)$	$\nu\Sigma_f(k)$	$X(k)$
1	1.466676	.01083982	.01648968	.00542511	1
2	0.364366	.09871243	-	.1305982	0

Region II: Westinghouse PWR fuel assemblies, 3% enrichment,
    10,000 MWD/T

Group(k)	$D(k)$	$\Sigma_a(k)$	$\Sigma_R(k)$	$\nu\Sigma_f(k)$	$X(k)$
1	1.461862	.01033693	.01694346	.00587230	1
2	0.366341	.09799408	-	.1357601	0

Reflector: pure water

Group(k)	$D(k)$	$\Sigma_a(k)$	$\Sigma_R(k)$	$\nu\Sigma_f(k)$	$X(k)$
1	1.136999	.0005892	.04830	0.0	-
2	0.149380	.0019240	-	0.0	-

lattice spacing = 10 cm for all regions

Control Poison Cross Sections:
    $\sigma_p(1)$ = 2.86715 cm^{-1} per gm of B^{10}/cc
    $\sigma_p(2)$ = 130.693 cm^{-1} per gm of B^{10}/cc

Materials interfaces occur at points 6 and 15.
Control points are those numbered 1 through 15.
To achieve regionwise poison concentrations:
    points 1 through 5 are constrained to the same value,
    points 7 through 14 are constrained to the same value,
    point 6 is constrained to the volume weighted average
        of the values in the two regions,
    point 15 is weighted by the fraction of its associated
        volume that is within Region II.
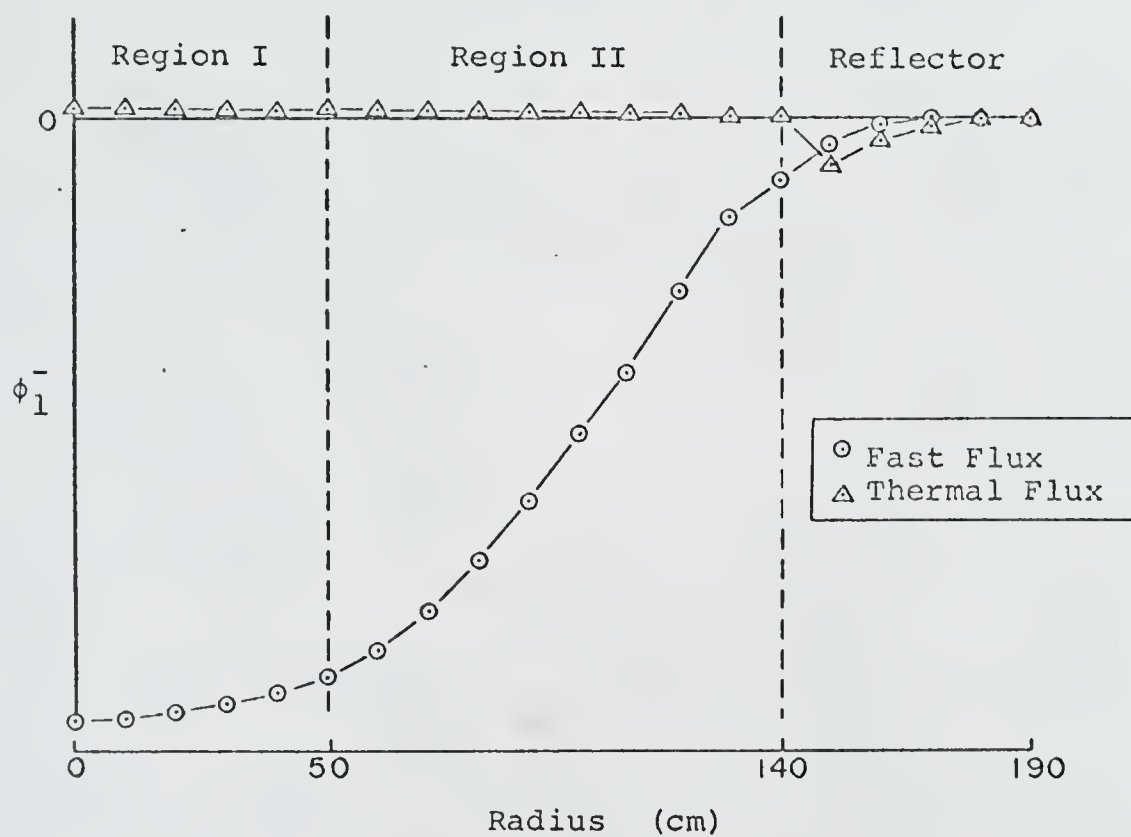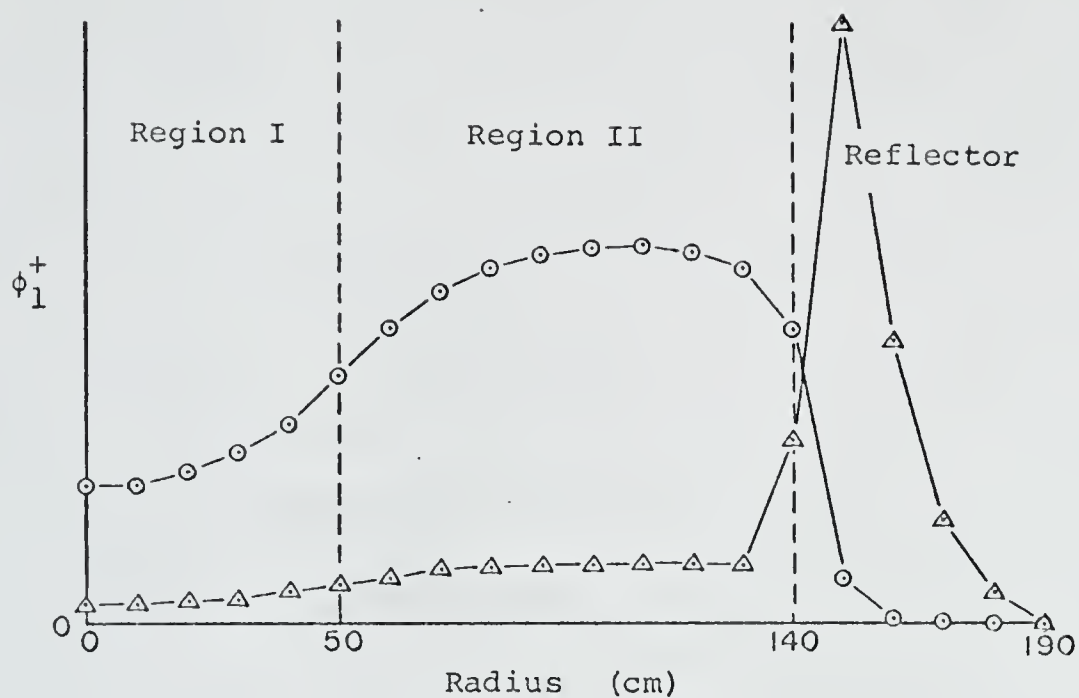
Figure 6   Poison Eigenvectors ϕ_1^+ and ϕ_1^- for Example 3
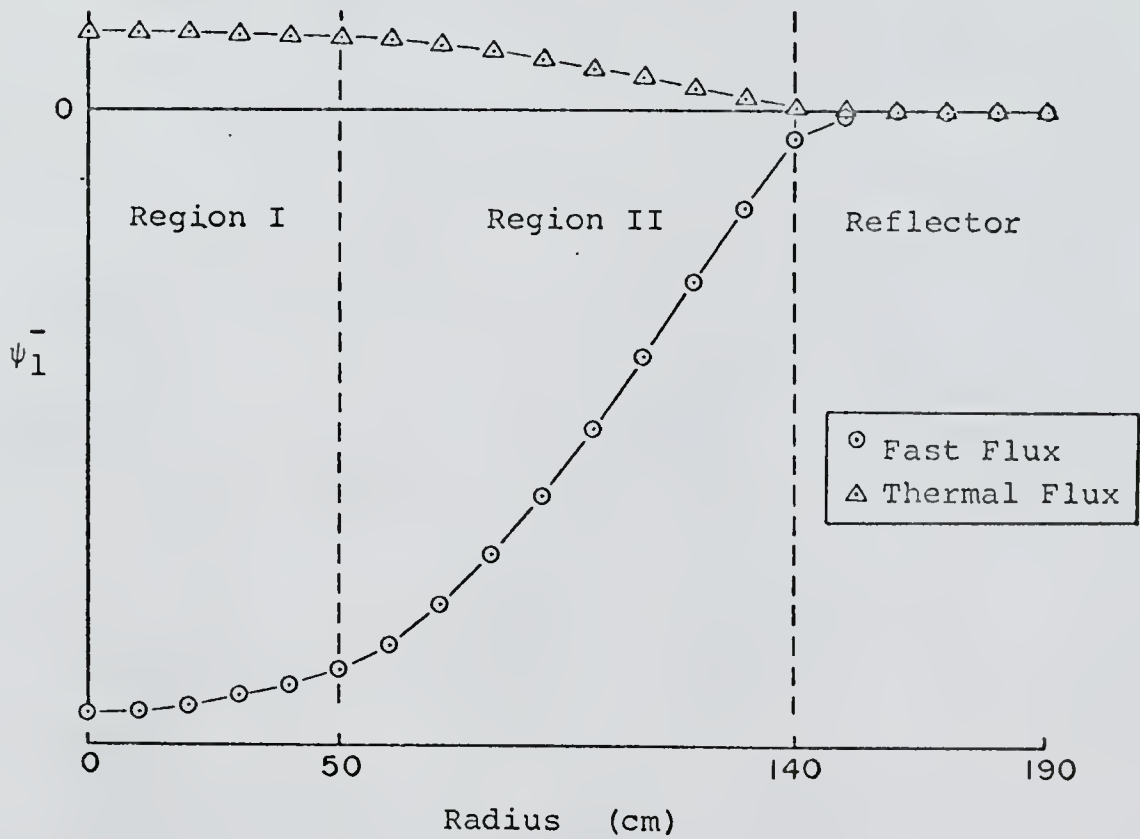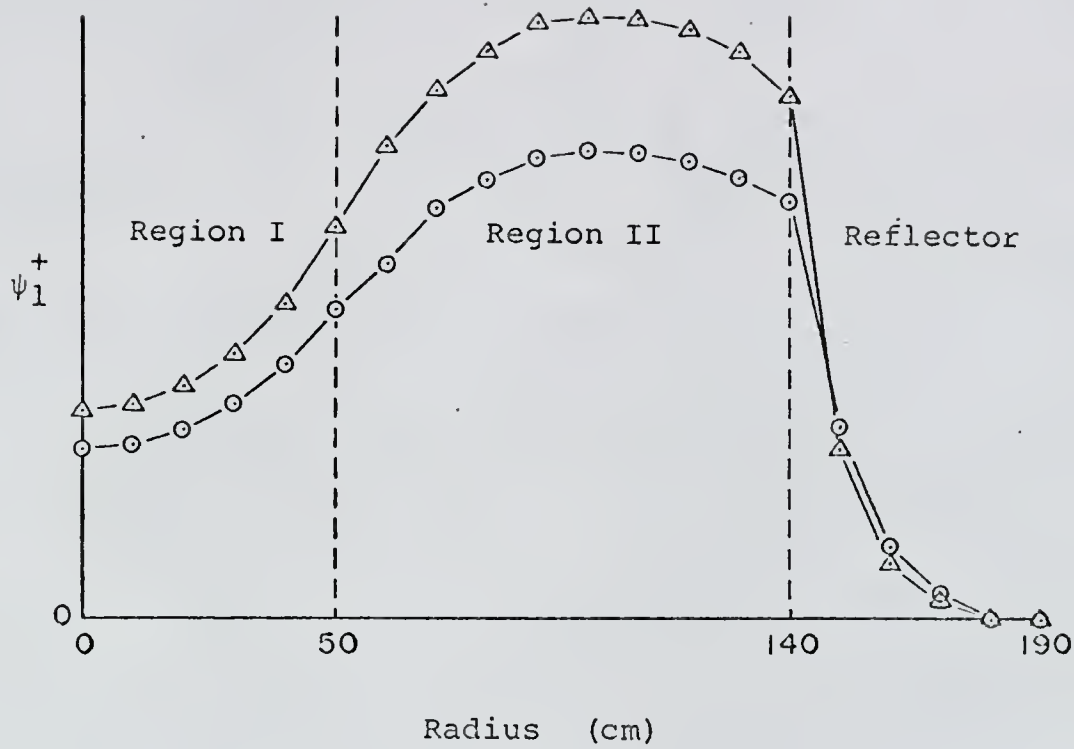
Figure 7   Adjoint Vectors ψ_1^+ and ψ_1^- for Example 3

modes are expected to be small, they may still be present.
In order to give the best opportunity for successful solu-
tion with partial expansions, the problem was designed so
that the interface between the two poison regions concides
with the interface between the two fueled regions. This
should allow the use to maximum advantage of any representa-
tion of an interface at that point which may already be
built into the poison eigenvectors. Except for the success
of the partial expansions, however, no evidence is offered
that there is indeed any advantage to such a coincidence
of interfaces.

Optimum two-region poison concentrations were calcu-
lated for Example 3 using complete and several partial
expansions of the problem. The resulting "optima" are
listed in Table 7 as a function of the number of vectors
used. As may be seen from these results, the two-region
optimum poison concentrations may be calculated with an accu-
racy of better than 5 percent using as few as five of the
thirty eigenvectors.

Table 8 lists the expansion coefficients found for the
optimum of each expansion. As can be seen, there is little
variation in the modes which give the most significant con-
tributions to the flux shape. This is in large part due
to the grossly underspecified nature of Example 3. Consi-
derable compensation by the remaining modes for those that
have been deleted will be seen when partial expansions of
the completely specified Example 1 are considered.

TABLE 7

Comparison of Reactor Configurations Resulting from the
Two-Region Poison Optimization of Example 3
Using Full and Partial Expansions

Expansion Designations:
   A   full expansion using complete set of 30 vectors,
       1+ through 15+ and 1- through 15-
   B   partial expansion using 20 vectors, 1+ through 10+
       and 1- through 10-
   C   partial expansion using 10 vectors, 1+ through 5+
       and 1- through 5-
   D   partial expansion using 5 vectors, 1+ through 5+

Solutions for Optimum Two-Region Poison Concentrations:

Region	Exp. A	Exp. B	Exp. C	Exp. D
I	-1.1740E-05	-1.1680E-05	-1.2306E-05	-1.2310E-05
II	6.1978E-05	6.1956E-05	6.1935E-05	6.1936E-05

Power Density as a Function of Position

Point	Exp. A	Exp. B	Exp. C	Exp. D
1	1.26999	1.28071	1.35300	1.35318
2	1.26407	1.26381	1.33234	1.33249
3	1.24639	1.23678	1.27517	1.27524
4	1.21716	1.22184	1.19734	1.19736
5	1.17589	1.17008	1.12347	1.12348
6-	1.09913	1.09822	1.09448	1.09452
6+	1.15214	1.15118	1.14727	1.14730
7	1.07622	1.07691	1.10928	1.10936
8	1.02445	1.02184	1.03187	1.03196
9	0.97418	0.97586	0.96187	0.96192
10	0.92416	0.92327	0.91062	0.91061
11	0.87396	0.87441	0.87134	0.87130
12	0.82344	0.82418	0.83118	0.83114
13	0.77352	0.77357	0.78294	0.78291
14	0.76215	0.76262	0.76405	0.76403
15-	1.94450	1.94757	1.93433	1.93423

Worst Error Relative To Full Expansion Solution:

Exp. A	Exp. B	Exp. C	Exp. D
0.0%	0.8%	6.5%	6.5%

TABLE 8

Comparison of the Expansion Coefficients for the
Full and Partial Expansion Optimums for Example 3

Vector	Exp. A	Exp. B	Exp. C	Exp. D
1+	1.0000E 00	1.0000E 00	1.0000E 00	1.0000E 00
1-	1.4556E-06	1.1959E-06	-3.0287E-06	-
2+	1.1753E 00	1.1720E 00	1.1718E 00	1.1720E 00
2-	-1.1252E-04	-1.1248E-04	-1.1686E-04	-
3+	1.6364E 00	1.6315E 00	1.6436E 00	1.6439E 00
3-	-1.4268E-04	-1.4215E-04	-1.3965E-04	-
4+	5.8640E-01	5.8470E-01	6.0665E-01	6.0677E-01
4-	2.7469E-05	2.7695E-05	3.2615E-05	-
5+	-8.1452E-02	-8.1125E-02	-5.7227E-02	-5.7219E-02
5-	1.4756E-04	1.4729E-04	1.4921E-04	-
6+	-2.4115E-01	-2.4048E-01	-	-
6-	8.6860E-05	8.6557E-05	-	-
7+	-1.1751E-01	-1.1740E-01	-	-
7-	-2.6699E-05	-2.6734E-05	-	-
8+	3.7470E-02	3.7069E-02	-	-
8-	-6.2476E-05	-6.2342E-05	-	-
9+	8.7342E-02	8.7095E-02	-	-
9-	-2.6800E-05	-2.6700E-05	-	-
10+	4.7406E-02	4.7924E-02	-	-
10-	1.1339E-05	1.1332E-05	-	-
11+	-7.5053E-03	-	-	-
11-	1.6482E-05	-	-	-
12+	-2.2150E-02	-	-	-
12-	5.5250E-06	-	-	-
13+	-1.0283E-02	-	-	-
13-	-4.2789E-07	-	-	-
14+	-1.6935E-02	-	-	-
14-	-2.7144E-06	-	-	-
15+	7.3939E-03	-	-	-
15-	2.6048E-12	-	-	-

Before leaving Example 3, it is instructive to examine the "optimum realizable power distribution" as computed by the GAUSS program. The shape found for the full expansion solution is plotted in Figure 8. This shape is "optimum" only in the weighted least squared errors sense. That is, it gives the minimum value of the error as computed by equation (86). This is the form required by the Gaussian optimization technique in order to predict an optimum, but it obviously is not representative of the actual limitations placed on the power distribution, or of the actual penalty associated with a power peak. In practice, it is the maximum power density which must be limited to some safe value, not the average. Any other peaks which come nearly to the same value only serve to further increase total power output. Therefore, one might suspect that the true physically optimum two-region poison distribution would result when the concentration in region I is decreased to the point that the central peak at point 1, or the interface peak at point 6, coincides with the "two-group peak" caused by the reflector at point 15.

Thus, an optimization technique utilizing a more realistic error function is certainly to be desired. The purpose of this study, however, has not been to select an optimization technique. It has been, rather, to introduce a powerful technique for the evaluation of the power distribution produced by each poison distribution specified by whatever optimization technique an investigator chooses
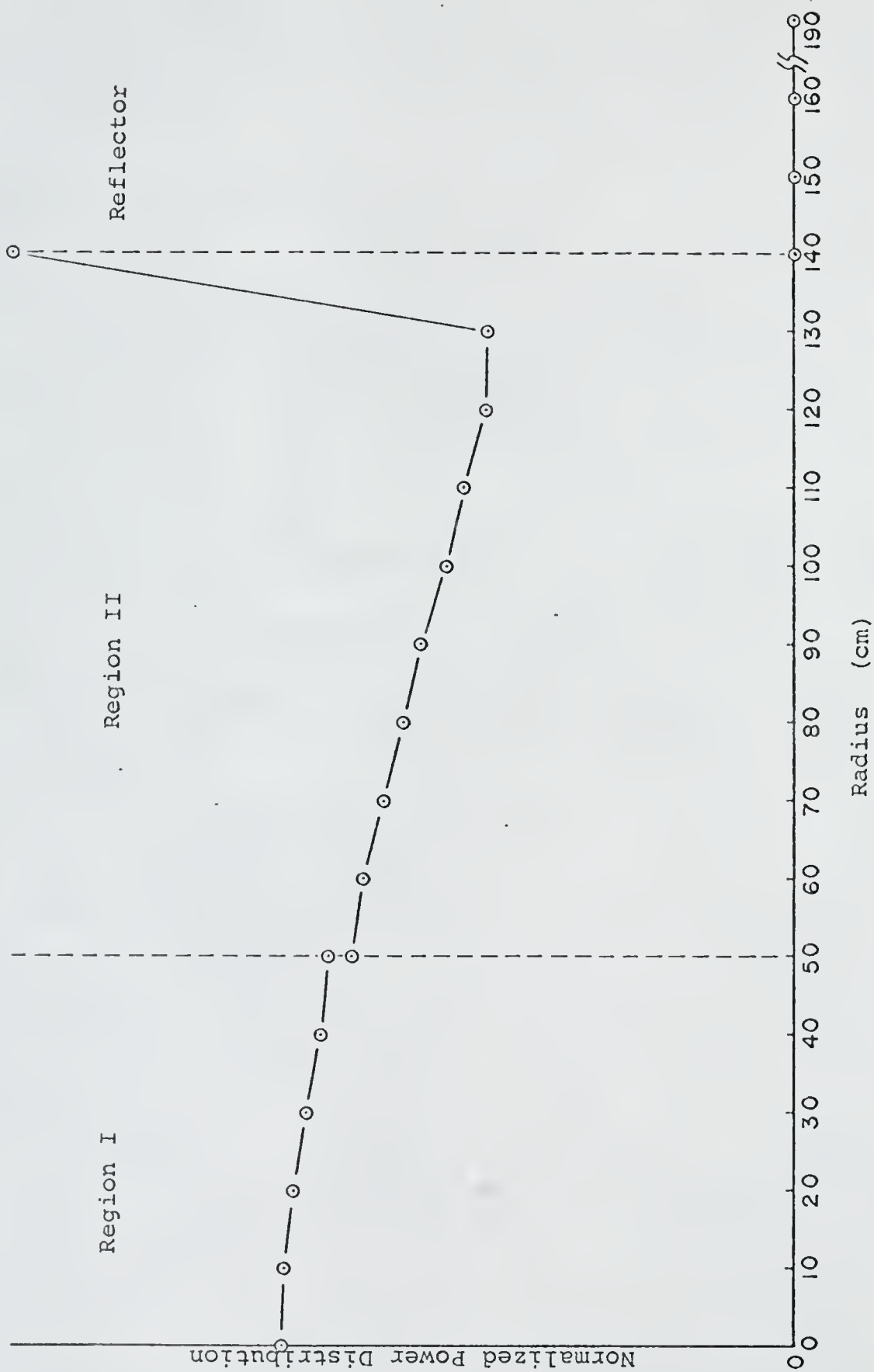
Figure 8   Power Distribution for the Optimum Two-Region Poison Distribution in Example 3

to use.   In that context, the GAUSS program provides as good a demonstration tool as would a technique providing a more realistic optimum, in that the flux-power construction methods employed are independent of the error function used.

In order to investigate the possibilities for partial expansions in completely specified problems, tests were made using the mathematical model of Example 1.   These tests are not the most general in nature, due to the separability of the energy and space variables and to the small number of points used to represent the space dimension. However, tests on a model utilizing a large number of control points would require prohibitively expensive amounts of computer time for the GAUSS program to independently manipulate the poison concentration at each control point. Also, inclusion of an interface between two fueled regions creates an inherently underspecified problem (assuming a mesh point is located on the interface.)   This is due to the discontinuity of the power density at such interfaces. Two values for the power density must be calculated for the point, while, at most, only one poison concentration may be specified to control them.   Example 1 seems sufficient at least to represent two important properties for partial expansions of completely specified problems.

Because the series of eigenvectors with negative fast/ thermal flux ratios played such a small part in the solution of the fully expanded problem, the necessity of their inclusion

in the expansion may seem in question. A solution was found for the problem as expanded using only the positive flux ratio eigenvectors, and the results are compared with the full solution in Table 9. Because the five vectors used are still sufficient to expand any power vector, it was still possible for the GAUSS program to find a set of poison concentrations which reduced the error function to zero. However, the resulting poison concentrations are significantly different, especially near the outer boundary, where the spectral shift is greatest from the undisturbed system. Still, these values are accurate within about 6 percent.

Even for completely specified problems, then, the elimination of the series of negative flux ratio eigenvectors leads to a reasonable approximation of the optimum poison concentrations. However, the examples considered in this study are special cases in a way that has not been mentioned as yet. They are all two-group representations of thermal reactors, and, as such, allow little variation in the energy spectrum of a critical reactor. All source neutrons appear in the fast group, and, due to its much larger cross section, most fissions are produced in the thermal group. In a fast reactor, the spectrum is not dominated by these phenomena, and introductions of poison material may cause gross spectral shifts. In such cases, elimination of the negative flux ratio eigenvectors may lead to extremely erroneous results.

Another phenomenon which may show up in partial expansions

TABLE 9

Results of Partial Expansions for a Completely
Specified System (Example 1)

Expansion Designations:

A   partial expansion using 5 vectors, ϕ_1^+ through ϕ_5^+

B   partial expansion using 4 vectors, ϕ_1^+ through ϕ_4^+

C   partial expansion using 3 vectors, ϕ_1^+ through ϕ_3^+

D   partial expansion using 8 vectors, ϕ_1^+ through ϕ_4^+

   and ϕ_1^- through ϕ_4^-

Solutions of Stopping Points* for Optimum Pointwise Poison
   Concentrations:

Point	Full	Exp. A	Exp. B
1	6.55390E-02	6.55969E-02	9.15413E-02
2	6.54151E-02	6.57080E-02	4.20902E-02
3	6.41074E-02	6.43932E-02	8.52438E-02
4	5.16011E-02	4.88118E-02	3.44118E-02
5	-1.02413E-01	-1.00152E-01	-9.30235E-02

Point	Exp. C	Exp. D
1	3.23415E-02	1.53594E-01
2	7.53454E-02	7.96075E-02
3	1.11435E-01	1.43480E-01
4	-2.34082E-01	6.90504E-02
5	-4.13993E-02	-1.69730E-01

*Searches for the solutions to expansions B, C and D
were terminated by the inability of the GAUSS
optimization routine to find meaningful derivatives.

is compensation by the remaining modal components for those that have been deleted. For instance, in a case where only the positive flux ratio eigenvectors have been used, except that for one vector, ϕ_i^- has been substituted for ϕ_i^+, the coefficient for ϕ_i^- would be greatly increased over its value in a full solution. This is because the shape, $\phi_i^{\pm}$, is necessary to flatten the power distribution, and it will be used by the GAUSS program, if available, no matter what happens to the spectrum of the flux. Obviously, an intelligent approach must be taken to providing a set of eigenvectors to be used in partial expansions. The vectors which are expected to give the smallest contributions to the solution should be eliminated first.

In addition to the change in the predicted optimum values, there is also a decrease in the ability of the GAUSS program to find this optimum as the number of vectors used in the expansion is decreased. This is due to the fact that the power calculated in the expanded equation no longer completely represents the power in the unexpanded formulation. The "error hypersurface" in "concentration space" is made increasingly more rippled, until many local minima result. It becomes increasingly likely that the GAUSS program will not be able to distinguish these false minima from the one associated with the physical optimum. Finally, the rippling of the error surface becomes so exaggerated that the optimization program can no longer calculate

meaningful derivatives for the prediction of the optimum point. In general, this phenomenon may be expected to become important sooner as the problem becomes more nearly completely specified.

Partial expansions of Example 1 were carried out for several sets of eigenvectors which are not capable of completely representing the "desired power vector". The results are also given in Table 9. These solutions were all terminated by the GAUSS program due to an inability to find meaningful derivatives, rather than due to the minimization criteria for the error value. As may be seen in the table, the expansions using the first three and four positive flux ratio eigenvectors provided poison distributions in the general vicinity of the true values. However, the expansion using the first four negative modes, as well as the first four positive modes, provided a much less satisfactory solution. Although it provided the same four shapes as the expansion using only four positive ratio eigenvectors, it provided twice as many coefficients to be found in solution of the expanded equations. This apparently gave more occasion for the coefficient compensation phenomenon, producing less valid representations of the flux and power than did the expansion using only positive flux ratio modes. The lesson to be drawn from this experience thus seems to be to eliminate the modes in the order that they contribute to the desired solution, from smallest contribution to largest. Since the solution is not known beforehand, this

method requires some experience and insight, as does any approximation technique.

The negative poison concentrations occassionally found in the solutions to these examples represent a problem encountered in any automated poison distribution search. If there is already some poison in the reactor, then the negative concentration may not exceed this magnitude. If there is no initial poison, then the optimum poison concentrations must be constrained to be positive. Because this study is directed at the evaluation of the effects of a trial poison distribution, once selected, rather than the actual selection of the distributions for trial, this problem will not be considered further.

There is one more phenomenon of the poison eigenvectors which the author yet wishes to demonstrate. Example 4 is included to illustrate the concepts advanced in Chapter V for the identification of the eigenvectors. The reactor model is a two-group, forty-four-point representation of a cylindrical reactor with three fueled regions and a water reflector. The reactor model is described in Table 10, and shown schematically in Figure 9. The poison eigenvectors ϕ_1^+ and ϕ_5^- are given in Table 11. All of the even numbered points within fueled regions are control points, giving rise to 32 poison eigenvectors. The fundamental mode, ϕ_1^+, is easily identified in that it is everywhere positive. However, without the concept of the region of completeness, it may

TABLE 10

Reactor Model for Example 4

Core Configuration: finite cylinder, three homogeneous
     fueled regions from radii 0 to 90 cm, 90 to 130 cm,
     and 130 to 160 cm, pure water reflector from radius
     140 to 220 cm, 366 cm core height
Mathematical Model: diffusion theory, 2 energy groups,
     44 space points, 16 control points, zero current
     condition at centerline (point 1), zero flux condition
     at extrapolated boundary (point 45), continuous
     neutron current conditions at internal interfaces
     (points 19, 27, and 33)

Materials Parameters (in standard metric units):

Region I: Westinghouse PWR fuel assemblies,
            3% enrichment, 18,000 MWD/T

Group(k)	$D(k)$	$\Sigma_a(k)$	$\Sigma_R(k)$	$\nu\Sigma_f(k)$	$X(k)$
1	1.466676	.01083982	.01648968	.00542411	1
2	0.364366	.09871243	-	.1305982	0

Region II: Westinghouse PWR fuel assemblies,
             3% enrichment, 10,000 MWD/T

Group(k)	$D(k)$	$\Sigma_a(k)$	$\Sigma_R(k)$	$\nu\Sigma_f(k)$	$X(k)$
1	1.461862	.01033693	.01694346	.00587230	1
2	0.366341	.09799408	-	.1357601	0

Region III: Westinghouse PWR fuel assemblies,
              3% enrichment, 0 MWD/T

Group(k)	$D(k)$	$\Sigma_a(k)$	$\Sigma_R(k)$	$\nu\Sigma_f(k)$	$X(k)$
1	1.455021	.00949965	.01768530	.00648183	1
2	0.374245	.08487368	-	.1307898	0

Reflector: pure water

Group(k)	$D(k)$	$\Sigma_a(k)$	$\Sigma_R(k)$	$\nu\Sigma_f(k)$	$X(k)$
1	1.136999	.0005892	.04830	0.0	-
2	0.149380	.0019240	-	0.0	-

lattice spacing = 5 cm for all regions

Control Poison Cross Sections:
     $\sigma_p(1)$ = 2.86715 cm^{-1} per gm of B^{10}/cc
     $\sigma_p(2)$ = 130.693 cm^{-1} per gm of B^{10}/cc

Control points are those with the even indices 2 through 32.

B_z^2 = 7.380E-05 cm^{-2}
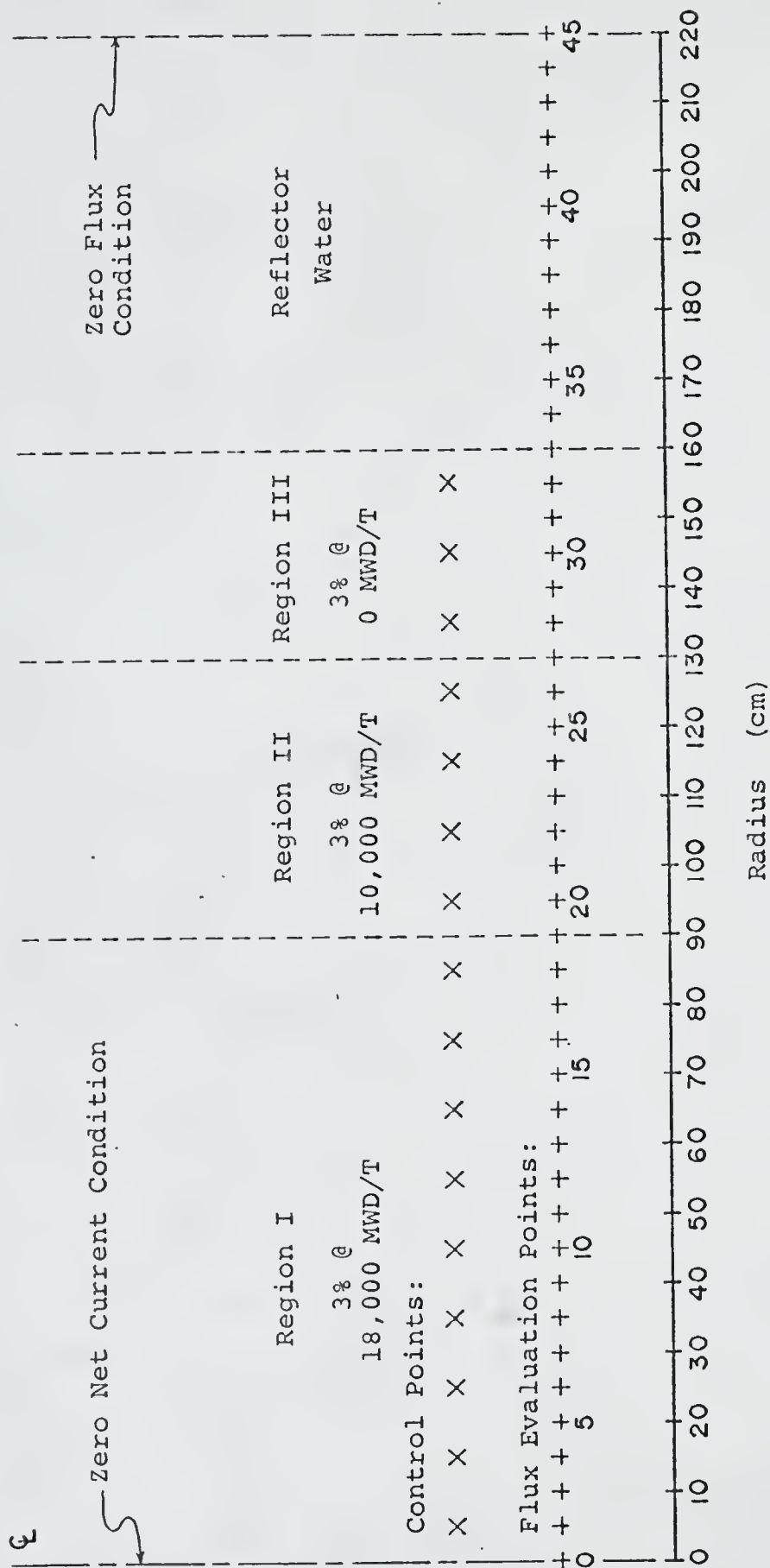
Figure 9   Schematic Representation of Reactor Model for Example 4

TABLE 11

The Poison Eigenvectors $\bar{\phi}_1^+$ and $\bar{\phi}_5^-$ for Example 4

Point No.	Material Region	Control Point	ϕ_1^+ Fast Flux	ϕ_1^+ Thermal Flux	ϕ_5^- Fast Flux	$\bar{\phi}_5^-$ Thermal Flux
1	I		6.48608E 00	1.00000E 00	9.80949E 00	1.00000E 00
2	I	x	6.53801E 00	8.59102E-01	1.01731E 01	-8.09185E-02
3	I		6.97182E 00	1.11013E 00	6.57654E 00	8.35440E-01
4	I	x	7.34792E 00	9.67144E-01	4.90493E 00	-3.93769E-02
5	I		8.21532E 00	1.30828E 00	1.40692E 00	1.78712E-01
6	I	x	9.04132E 00	1.19004E 00	-1.19811E 00	9.61844E-03
7	I		1.04985E 01	1.67188E 00	-2.73261E 00	-3.47106E-01
8	I	x	1.19323E 01	1.57057E 00	-4.26426E 00	3.42333E-02
9	I		1.42338E 01	2.26672E 00	-3.41058E 00	-4.33224E-01
10	I	x	1.65421E 01	2.17732E 00	-2.95414E 00	2.37157E-02
11	I		2.00972E 01	3.20047E 00	-1.01203E 00	-1.28551E-01
12	I	x	2.37105E 01	3.12084E 00	6.56004E-01	-5.26637E-03
13	I		2.91656E 01	4.64461E 00	1.83511E 00	2.33103E-01
14	I	x	3.47647E 01	4.57532E 00	3.08280E 00	-2.47485E-02
15	I		4.31314E 01	6.86867E 00	2.61119E 00	3.31683E-01
16	I	x	5.17843E 01	6.81622E 00	2.40523E 00	-1.93091E-02
17	I		6.46424E 01	1.02968E 01	9.36960E-01	1.19017E-01
18	I	x	7.80149E 01	1.02212E 01	-3.58320E-01	2.89439E-03
19	I/II		9.77787E 01	1.58216E 01	-1.38886E 00	-1.79251E-01
20	II	x	1.16870E 02	1.58808E 01	-2.47249E 00	2.03839E-02
21	II		1.42561E 02	2.33968E 01	-2.15837E 00	-2.82985E-01
22	II	x	1.65201E 02	2.23902E 01	-1.99341E 00	1.64480E-02
23	II		1.97467E 02	3.23075E 01	-7.51408E-01	-9.85167E-02
24	II	x	2.25796E 02	3.06477E 01	3.87705E-01	-3.20718E-03
25	II		2.67532E 02	4.39759E 01	1.26836E 00	1.66280E-01
26	II	x	3.04048E 02	4.16790E 01	2.19111E 00	-1.81955E-02

Table 11 (continued)

Point No.	Material Region	Control Point	ϕ_1^+ Fast Flux	ϕ_1^+ Thermal Flux	ϕ_5^- Fast Flux	ϕ_5^- Thermal Flux
27	II/III		3.57785E 02	6.39107E 01	1.88118E 00	2.65215E-01
28	III	x	3.93799E 02	6.15990E 01	1.65683E 00	-1.43134E-02
29	III		4.35845E 02	8.43197E 01	5.10679E-01	7.74891E-02
30	III	x	4.43365E 02	7.01990E 01	-5.86943E-01	5.10908E-03
31	III		4.57036E 02	9.13105E 01	-1.31758E-01	-1.99064E-01
32	III	x	4.29169E 02	9.00181E 01	-2.04766E 00	2.40469E-02
33	III/Ref.		4.59593E 02	2.95419E 02	-1.61659E 00	-1.23365E 00
34	Ref.		1.30808E 02	9.29949E 02	-5.88063E 01	-4.12202E 00
35	Ref.		4.76051E 01	8.00604E 02	-2.14014E-01	-3.56624E 00
36	Ref.		1.73323E 01	5.51286E 02	-7.79195E-02	-2.45983E 00
37	Ref.		6.31297E 00	3.46875E 02	-2.83807E-02	-1.54899E 00
38	Ref.		2.30025E 00	2.08527E 02	-1.03410E-01	-9.31596E-01
39	Ref.		8.38438E-01	1.22151E 02	-3.76929E-03	-5.45851E-01
40	Ref.		3.05702E-01	7.03745E 01	-1.37432E-03	-3.14525E-01
41	Ref.		1.11469E-01	3.99702E 01	-5.01123E-04	-1.78656E-01
42	Ref.		4.05808E-01	2.22108E 01	-1.82435E-04	-9.92824E-02
43	Ref.		1.45717E-02	1.16615E 01	-6.55089E-05	-5.21289E-02
44	Ref.		4.68092E-03	4.97616E 00	-2.10436E-05	-2.22445E-02

be difficult to identify ϕ_5^-. If only the flux values at
the control points are considered, however, it may be
easily verified from Table 11 that these components exhibit
five sign changes in each group and a negative fast/thermal
flux ratio everywhere. The vector is thus readily identi-
fied as ϕ_5^-.

Finally, some mention should be made of the accuracy
possible in using the eigenvector expansion technique. The
eigenvectors and adjoint vectors generated for use in
Examples 1 through 4 have generally satisfied the ortho-
gonality condition very well. The cross products $\psi_m^T[\sigma_p]\phi_i$
result in values of 10^{-6} to 10^{-10} when $m \neq i$. The use of
M eigenvectors for a complete expansion may, the author
estimates, introduce an additional fractional error on the
order of $\sqrt{M} \times 10^{-6}$, which is usually rather small compared
to the errors associated with the materials parameters and
the problem formulation in descrete point, descrete energy
group diffusion theory. In using partial expansions, the
error will obviously be significantly greater. The accuracy
in such cases may only be gauged by approaching the observed
minimum from several directions, to insure that it is indeed
the optimum, and by increasing the number of vectors in
the expansion to see how this affects the location of the
minimum.

CHAPTER VIII

CONCLUSIONS AND SUGGESTIONS FOR FURTHER STUDY

In this study, it has been shown that the specification of a physically realizable power distribution gives no a priori insight as to the spectrum of the multigroup fluxes which must produce it. The spectrum must therefore be considered as a set of unknown functions. The spatial distribution of some control material (with known energy dependence of its cross section) must be found which maintains the desired power shape in the critical state. The problem is completely specified, leaving no freedom for arbitrary selection of some of the unknown functions, and a solution is known to exist from physical considerations. Unfortunately, the multigroup criticality equations for the controlled reactor are nonlinear. The control terms contain the products of the unknown poison distribution and the unknown spectrum functions. This nonlinearity has prevented the author from obtaining an analytic solution to the general problem, and forced him to consider the approach of using numerical optimization techniques to find solutions to particular problems.

The author has introduced a method which he believes will be of interest and utility to those who seek to optimize

reactor power distributions through poison management. In comparison to the conventional method, this method considerably reduces the amount of work necessary for adjusting the magnitude of a poison distribution to achieve criticality and for evaluating the resulting power distribution. It is this type of calculation which must be done repeatedly, for different poison distributions in the same reactor, during numerical searches for the optimum poison distribution.

This technique may be considered in two parts. First is the poison eigenvalue formulation of the criticality equation and a suitable transformation technique to ensure convergence to the proper eigenvalue. This formulation allows for the simultaneous adjustment of the magnitude of the poison distribution to the critical value, and the calculation of the flux shape in the critical reactor. Only one solution of the criticality equations is necessary for the effects of each poison distribution to be investigated. This is a considerable simplification over the conventional, reactivity eigenvalue formulation, where the criticality equations must be solved many times in order to adjust the magnitude of a poison distribution to achieve criticality. An additional advantage of the poison eigenvalue formulism is that it separates the control terms from the rest of the terms in the criticality equation. In this formulation, it is the matrix containing the noncontrol terms which must

be inverted in order to perform the power iteration for the flux. Since this matrix remains unchanged no matter how the control terms are varied, only one matrix inversion is necessary no matter how many poison distributions are investigated. (Improper choice of the method for ensuring convergence to the proper eigenvalue may destroy this property, however.) In the conventional, reactivity eigenvalue formulation, the control and some noncontrol terms are mixed on the same side of the criticality equation, and a matrix inversion (or equivalent) is necessary each time a new flux vector must be found.

Two transformation methods have been discussed which ensure convergence to the proper poison eigenvalue. One of these, the Wielandt transformation, has been demonstrated successfully by numerical examples. It was chosen for demonstration purposes because it is easily programmed. However, it has the undesirable feature of mixing control and noncontrol terms on the side of the criticality equation which must be inverted. A matrix inversion is thus required each time a new flux vector is found. The other method, the β_s transformation, has been advanced as offering similar benefits without the penalty of additional matrix inversions. Because of the complexity of the programming required to overcome its inherently slow convergence rate, this latter technique was not demonstrated.

The second part of the technique involves the expansion

of the criticality equation in a set of eigenvectors which
individually satisfy the boundary and interface conditions
required of the flux in a particular reactor. This set
of eigenvectors must be generated for each reactor for
which the optimum poison distribution is to be found. They
are then used to repeatedly expand the criticality equation,
as different poison distributions are added to the uncon-
trolled materials configuration.

The orthogonality relations and regions of completeness
were investigated for various set of eigenvectors, which
satisfy various eigenvalue equations associated with the
criticality equation. The fission eigenvectors where shown
to be incomplete in the energy dimension, and thus unsuitable
for the expansion of a general multigroup flux vector. The
poison eigenvectors were introduced and shown to be com-
plete in energy and space for the flux elements at control
points, only. It was then proven that this vector set is
capable of representing the flux at all points in the reac-
tor for any critical configuration which may be produced
by the poison material at the control points.

The poison eigenvectors were chosen as the set to be
used for the expansion, due to the great simplifications that
result from their use. The first of these simplifications
is the reduction of the order of the matrices in the criti-
cality equation. The unexpanded problem has an order given
by the product of the number of spatial points and the

number of energy groups used to model the reactor. The
poison eigenvector expansion reduces this order to the pro-
duct of the number of control points and the number of energy
groups. In addition, the matrix which contains all but the
control terms of the criticality equation is reduced to
diagonal form. This is true for all reactor models, whether
they employ one-dimensional or multidimensional represen-
tations of the spatial coordinates. Because it is this
matrix which must be inverted in order to perform the power
iteration solution for the flux, its reduction to dia-
gonal form has the advantage of making this inversion trivial.
In essence, the generation of the poison eigenvectors and
their adjoints has also provided for the inversion of this
matrix. The use of the β_s transformation, rather than the
Wielandt transformation technique, to ensure convergence
to the proper poison eigenvalue, will preserve the diagonal
nature of this matrix in the expanded equation. These sim-
plifications are possible because the poison eigenvectors
are a complete set which satisfy the criticality equation,
reflecting the geometry, the various regions of different
materials, the energy dependence of the poison cross sec-
tions and the positions at which poison may be introduced.

The technique, as a whole has been demonstrated by one-
dimensional, two-group numerical examples, and the Gaussian
least squares optimization method has been found capable of
handling the numerical search. These examples have shown
the feasibility of the method and allowed comparison of

flux shapes and reactivities with calculations made by a standard diffusion theory code. Agreement with the standard code was excellent for these parameters. In addition, these examples verified the necessity of using a set of eigen-vectors which are capable of expanding both the energy and space dimensions of the flux values at the control points. In cases where the most general variations of poison concen-trations are to be allowed, it was shown that all of the eigenvectors gave significant contributions to the optimum flux shape. In cases where the concentrations are to be somewhat constrained, for instance to regionwise variation, the use of relatively few eigenvectors in the expansion has led to quite reasonable approximations to the optimum poison distributions.

The method, as presented in this study, offers consi-derable power for the solution of the general problem, while at the same time allowing considerable simplification for approximating the less troublesome cases. Before this technique may be considered a practical tool for the reactor engineer, however, it will be necessary to embody these concepts in a generalized, well-programmed and well-docu-mented computer code. Toward that end, the author would like to propose the following areas for investigation. First, the β_s transformation technique should be programmed and tested for reliability and speed. If it can be made to perform as expected, it should be substituted for the

Wielandt transformation technique in ensuring convergence to the fundamental mode. In addition, a study should be made of the applications of various numerical optimization methods to this particular approach for choosing poison concentrations in a reactor. The goal should be to choose the method exhibiting good compatability in handling the variables, a form for the error function which most closely approximates the physical situation, and, of course, an approach sufficient to handle the nonlinearity so as to always obtain a solution. Included in this study must be the selection of some means for limiting the search to physically realizable values for the poison concentrations. Certainly, no optimization method will be perfect in all respects. However, the proper optimization method, incorporated in a code designed specifically for applying the techniques discussed in this study, should be a quite powerful tool in the hands of the reactor engineer.

The utility of such a code will, of course, depend upon the availability of easily handled, well-documented codes for the generation of multigroup, one or more dimensional poison eigenvectors for various reactor geometries. The principles for writing such codes are well in hand, today, and the growing abundance of time sharing, remote computer terminals should offer ample opportunity for using the on-line operator interaction techniques advocated in this study.

Another facet of the overall technique may be of interest when considering partially expanded problems with

regionwise poison variation. This involves using an approximation to the set of optimum concentrations, resulting from the solution of a partial expansion, to generate a new set of eigenvectors. By replacing the constant distribution with this "approximately optimum" distribution in generating the vectors, it may be possible to obtain a new set which is better able to represent the true optimum with the same order of partial expansion. In this manner, two sequential partial expansions may prove to offer better accuracy with less effort than would a single partial expansion using twice as many vectors. The total number of eigenvectors found would be identical in both cases. However, the order of the matrix equations to be solved in evaluating each poison distribution would be only half as great for the sequential expansion case. This approach also allows for an intuitive approximation of the optimum distribution to be used, in place of results from a previous partial expansion. Sufficiently accurate intuition may thus save even more work. It must be noted, however, that the parameters of the numerical search would no longer be true concentrations. They would be a set of pointwise multipliers for changing the concentrations used in generating the new set of eigenvectors.

This and other specializations of the general technique may be worthy of investigation for developing the full power of the method for use in various special classes of the poison distribution problem.

APPENDICES

APPENDIX I

A DESCRIPTION OF THE TECHNIQUES EMPLOYED
BY THE EIGENVECTOR GENERATING PROGRAM "CODE2"

This computer code was written purely for demonstrational purposes, and is therefore much less general and sophisticated than would be desirable for a standardized code that is designed to be an engineering tool. The program is limited to the calculation of two-group, one-dimensional poison eigenvectors in axisymmetric cylindrical geometry using diffusion theory. Mesh points are placed on the axis, at materials interfaces, at the extrapolated boundary, and at any number of points inbetween. The interval between mesh points must be constant within each material region. Each region is considered to be of homogeneous composition. Control points, as defined in Chapter IV, may be specified at any or all mesh points, with the exception of the extrapolated boundary point, for which flux values are never explicity calculated. The zero flux condition is used for the extrapolated boundary point, and the zero net current condition (cylindrical axial symmetry) is used at the axis point. Finite axial height of the reactor may be approximated by the specification of a constant axial buckling, B_z^2.

The derivation of the finite difference approximation

from the multigroup analytic equations is well documented in the literature.[24] The derivative therefore will not be repeated here. The appropriate equations for the general problem outlined above, written for point j located at the interface between materials regions I and II, are

$$-C(1,j,j+1)\phi(1,j+1)+C(1,j,j)\phi(1,j)$$

$$-C(1,j,j-1)\phi(1,j-1)-F(1,j)\phi(2,j)$$

$$= \beta G(1,j)\phi(1,j) \tag{110a}$$

for the fast group, and, for the thermal group,

$$-C(2,j,j+1)\phi(2,j+1)+C(2,j,j)\phi(2,j)-C(2,j,j-1)\phi(1,j-1)$$

$$-F(2,j)\phi(1,j) = \beta G(2,j)\phi(2,j), \tag{110b}$$

where

$$C(k,j,j-1) = \pi D_I^k[r(j-1)+r(j)]/[r(j)-r(j-1)],$$

$$C(k,j,j+1) = \pi D_{II}^k[r(j+1)+r(j)]/[r(j+1)-r(j)],$$

$$C(k,j,j) = C(k,j,j-1)+C(k,j,j+1)$$

$$+V_I(j)(\Sigma_R^k+B_z^2 D^k-\chi^k \nu\Sigma_f^k)_I$$

$$+V_{II}(j)(\Sigma_R^k+B_z^2 D^k-\chi^k \nu\Sigma_f^k)_{II},$$

$$V_I(j) = \pi[3r(j)^2-2r(j)r(j-1)-r(j-1)^2]/4$$

$$V_{II}(j) = \pi[r(j+1)^2+2r(j)r(j+1)-3r(j)^2]/4,$$

$$F(1,j) = V_I(j)[\nu\Sigma_f^2]_I + V_{II}(j)[\nu\Sigma_f^2]_{II},$$

$$F(2,j) = V_I(j)[\Sigma_R^1]_I + V_{II}(j)[\Sigma_R^1]_{II},$$

$$G(k,j) = \sigma_p^k(j)[V_I + V_{II}].$$

In this formulation, the reaction rates are equivalent to what would occur if the flux at point j was the value for the flux everywhere in the cell about mesh point j.

Wachspress[21] introduced a technique for solving group diffusion equations which involved writing the equations (110) in "supermatrix" form, where the elements of the supermatrix are themselves matrices. Because this super- matrix is tridiagonal for one-dimensional problems, the Choleski algorithm[24], extended to matrices with matrix elements, may be used in the power series iterations in place of direct matrix inversion.

The supermatrix notation of Wachspress has the form

$$
\begin{vmatrix}
[U1] & -[Y1] & [0] & \\
-[W2] & [U2] & -[Y2] & \dots \\
[0] & -[W3] & [U3] & \dots \\
[0] & [0] & -[W4] & \dots \\
. & . & . & \dots
\end{vmatrix}
\begin{vmatrix}
\overline{\phi 1} \\
\overline{\phi 2} \\
\overline{\phi 3} \\
\overline{\phi 4} \\
.
\end{vmatrix}
= \beta
\begin{vmatrix}
[G1] & [0] & \dots \\
[0] & [G2] & \dots \\
. & . & \dots \\
. & . & \dots \\
. & . & \dots
\end{vmatrix}
\begin{vmatrix}
\overline{\phi 1} \\
\overline{\phi 2} \\
\overline{\phi 3} \\
\overline{\phi 4} \\
.
\end{vmatrix} ,
$$

$$(111)$$

where

$$
[Uj] = \begin{vmatrix}
C(1,j,j) & -F(1,j) \\
-F(2,j) & C(2,j,j)
\end{vmatrix} ,
$$

$$[Wj] = \begin{vmatrix} C(1,j,j-1) & 0 \\ 0 & C(2,j,j-1) \end{vmatrix},$$

$$[Yj] = \begin{vmatrix} C(1,j,j+1) & 0 \\ 0 & C(2,j,j+1) \end{vmatrix},$$

$$[Gj] = \begin{vmatrix} G(1,j) & 0 \\ 0 & G(2,j) \end{vmatrix},$$

$$\overline{\phi j} = \begin{vmatrix} \phi(1,j) \\ \phi(2,j) \end{vmatrix}.$$

A Wielandt transformation applied to this formulation does not change the structure. It only replaces the [Uj] matrix with the matrix $[Uj] - \beta_g [Gj]$.

Applying the Choleski algorithm to the tridiagonal supermatrix, the supermatrix is factored into upper and lower triangular forms.

$$\begin{vmatrix} [X1] & [0] & [0]... \\ -[W1] & [X2] & [0]... \\ [0] & -[W2] & [X3]... \\ \cdot & \cdot & \cdot ... \end{vmatrix} \begin{vmatrix} [I] & -[Z1] & [0]... \\ [0] & [I] & -[Z2]... \\ [0] & [0] & [0]... \\ \cdot & \cdot & \cdot ... \end{vmatrix} \begin{vmatrix} \overline{\phi 1} \\ \overline{\phi 2} \\ \overline{\phi 3} \\ \cdot \cdot \end{vmatrix}$$

$$= \beta \begin{vmatrix} [G1] & [0] & [0]... \\ [0] & [G2] & [0]... \\ [0] & [0] & [G3]... \\ \cdot & \cdot & \cdot ... \end{vmatrix} \begin{vmatrix} \overline{\phi 1} \\ \overline{\phi 2} \\ \overline{\phi 3} \\ \cdot \cdot \end{vmatrix}, \qquad (112)$$

where

$$[Zj] = \{[Uj] - [Wj][Zj-1]\}^{-1}[Yj], \qquad (113)$$

$$[Xj] = [Yj]\{[Zj]\}^{-1} = [Uj] - [Wj][Zj-1]. \qquad (114)$$

The index j in the above two series may be used from 1 to J (the maximum number of space points) if the definitions are made that $[A1] = [0]$ and $[CJ] = [I]$.

When equation (112) is used for iteration purposes, the known starting vector is substituted for the flux vector on the right hand side to begin the iteration. Thus, the left hand side becomes a known supervecter with vector elements, $\overline{Vj}$. The flux vector on the right hand side of equation (112) becomes the once iterated supervector with elements $\overline{Vj}'$. If another (unknown) supervector, with vector elements $\overline{Tj}$, is defined by the equation

$$
\begin{vmatrix}
[I] & -[Z1] & [\ 0\ ]... \\
[0] & [\ I\ ] & -[Z2]... \\
[0] & [\ 0\ ] & [\ I\ ]... \\
\cdot & \cdot & \cdot\ ...
\end{vmatrix}
\begin{vmatrix}
\overline{T1} \\
\overline{T2} \\
\overline{T3} \\
\cdot
\end{vmatrix}
=
\begin{vmatrix}
\overline{V1}' \\
\overline{V2}' \\
\overline{V3}' \\
\cdot
\end{vmatrix},
\qquad (115)
$$

then the elements of this vector may be found by substitution of equation (115) in equation (112). This gives rise to an algorithm for the elements $\overline{Tj}$.

$$
\overline{Tj} = [Xj]^{-1}\{\overline{Vj} - [Wj]\overline{Tj-1}\} \qquad (116)
$$

A more convenient algorithm is found by substituting from equation (114) for $[Xj]^{-1}$.

$$
\overline{Tj} = [Zj][Yj]^{-1}\{\overline{Vj} - [Wj]\overline{Tj-1}\} \qquad (117)
$$

Since $[Yj]$ is diagonal, inversion is trivial. With the $\overline{Tj}$

now known, the $\overline{Vj}'$ may be found from equation (115) by the algorithm

$$\overline{VJ}' = \overline{TJ} \quad \text{and} \quad Vj = Tj + [Zj]\overline{Tj-1}. \tag{118}$$

Thus, the new vector may be found by applying first algorithm (117) and then algorithm (118). The only matrix inversion necessary is the inversion of the two by two matrices $\{[Uj]-[Wj][Zj-1]\}$ when finding the $[Zj]$ matrices from equation (113). In general, for a K-energy-group formulation, only K by K matrices must be inverted, and this need be done only once per set of iterations to find one mode.

Once the new vector is found, it is normalized to unit magnitude and the iteration is performed repeatedly until only the dominant mode remains, as discussed in Chapter V.

The CODE2 computer program, which uses these techniques to find the poison eigenvectors, is listed in the next appendix. It is written in the Fortran II language for use on the IBM 1800 computer. This machine was chosen because of its availability for online human interaction with the calculations while they are in progress. The desire for such capability is explained in Chapter V.

APPENDIX II

LISTING OF THE DEMONSTRATION POISON EIGENVECTOR
GENERATION PROGRAM "CODE2"

Partial Dictionary to the Computer Code Variables:

<u>CODE2 Variables</u>   <u>Name or Variable in Text</u>

CODE2 Variables	Name or Variable in Text
A1(J)	fast elements of the $[W_j]$ matrices
A2(j)	thermal elements of the $[W_j]$ matrices
B(M,N,J)	$[U_j]$ matrices
BZS	B_z^2
C1(J)	fast elements of the $[Y_j]$ matrices
C2(J)	thermal elements of the $[Y_j]$ matrices
EV	ε
F1(J)	fast flux vector
F2(J)	thermal flux vector
FA1(J)	fast adjoint vector
FA2(J)	thermal adjoint vector
G1(J)	fast elements of the $[G_j]$ matrices
G2(J)	thermal elements of the $[G_j]$ matrices
GAM(M,N,J)	$[Z_j]$ matrices
INTER(I)	list of interface indices
NC(I)	list of control point indices
NCP	number of control points
NP	number of flux points
NR	number of regions
O1(J)	fast elements of the $\overline{T_j}$ vectors
O2(J)	thermal elements of the $\overline{T_j}$ vectors
RAD(J)	list of radii at flux points
SP1	σ_p^1
SP2	σ_p^2
W(M,N)	working 2x2 matrix
X1(J)	working vector for fast components
X2(J)	working vector for thermal components

```
C     THIS PROGRAM SOLVES THE TWO GROUP POISON EIGENVALUE PROBLEM IN CYLINDRICAL
C     GEOMETRY.  DIFFUSION THEORY IS USED, AND THE CRITICALITY EQUATION IS
C     WRITTEN IN SUPERMATRIX FORM, WITH THE FLUX VECTOR BEING COMPOSED OF THE FAST
C     AND THEN THE THERMAL FLUXES AT EACH POINT IN SUCCESSION.  A GUESS IS MADE
C     AT THE EIGENVALUE OF THE DESIRED MODE, AND THE PROGRAM PERFORMS A WIELANDT
C     TRANSFORMATION OF THE CRITICALITY PROBLEM, USING THIS GUESS.  THE EIGEN-
C     VALUE AND VECTOR ARE THEN OBTAINED BY THE POWER ITERATION TECHNIQUE.
C     THE TRIDIAGONAL SUPERMATRIX (WITH MATRIX ELEMENTS) IS INVERTED
C     BY USING THE CHOLESKI ALGORITHM.  THE GENERAL INVERSE OF THE 2X2 "ELEMENTS"
C     ON THE DIAGONAL OF THE SUPERMATRIX ARE COMPUTED IN SUBROUTINE QINV.
C         SENSE SWITCH 0 (WHEN REQUESTED IN PRINTOUT) ALLOWS ANOTHER
C              EIGENVALUE GUESS TO BE MADE FOR THE SAME PROBLEM.
C         SENSE SWITCH 1 (WHEN REQUESTED IN PRINTOUT) ALLOWS ANOTHER
C              CORE GEOMETRY PROBLEM TO BE RUN.
C         SENSE SWITCH 2 TERMINATES VECTOR PRINTOUT.
C         SENSE SWITCH 3 ALLOWS ITERATION FOR THE ADJOINT VECTOR EVEN
C              IF THE EIGENVECTOR DID NOT CONVERGE.
C         DATA SWITCH 0 GIVES A FAST/THERMAL FLUX RATIO OF -1 FOR THE
C              STARTING VECTORS IN THE ITERATIONS.
C         TYPEWRITTEN "YES" OR "NO" WILL ANSWER QUESTION IN PRINTOUT
C              ABOUT PUNCHING THE VECTORS ONTO CARDS.
      DIMENSION A1(50),A2(50),B(2,2,50),    C1(50),C2(50),GAM(2,2,50),
     1  G1(50),G2(50),INTER(50),NC(50),RAD(50),O1(50),O2(50)
      DIMENSION W(2,2),F1(50),F2(50),FA1(50),FA2(50),X1(50),X2(50)
      WRITE(1,148)
  148 FORMAT('POSITION PAPER FOR PRINTOUT AND PUSH START BUTTON')
      PAUSE 1111
   71 CONTINUE
C     SET UP PROBLEM GEOMETRY
      WRITE(1,501)
  501 FORMAT(//'PROGRAM CODE2')
      READ (2,100) NP,NR,NCP
  100 FORMAT(1615)
      NPM = NP-1
      NRP = NR+1
      READ (2,100) (INTER(I),I=2,NRP)
```

```
      INTER(1) = 1
      DO 47 I=NCP,50
   47 NC(I) = 0
      READ (2,100)  (NC(I),I=1,NCP)
      READ (2,101)  SP1,SP2,BZS
  101 FORMAT(8F10.0)
      WRITE(1,502) NP,NR,NCP
  502 FORMAT(//'REACTOR MODEL HAS'I3,' POINTS IN'I3,' REGIONS, WITH'I3,
     1' CONTROL POINTS.')
      WRITE(1,503) (INTER(I),I=1,NRP)
  503 FORMAT(/'INTERFACES OCCUR AT POINTS NO:'/(20I5))
      WRITE(1,504) (NC(I),I=1,NCP)
  504 FORMAT(/'CONTROL POINTS ARE THOSE NUMBERED:'/(20I5))
      WRITE(1,505) BZS
  505 FORMAT(/'AXIAL BUCKLING ='E13.6,'/CM2.')
      WRITE(1,506) SP1, SP2
  506 FORMAT(/'POISON CROSS SECTIONS:    FAST ='E13.6,'/CM,    THERMAL ='
     1 E13.6,'/CM.')
      WRITE(1,507)
  507 FORMAT(//'MATERIALS PARAMETERS   (FAST/THERMAL):'/13X,'DELTA R'5X,
     1'DIFFUSION COEF.'5X,'ABSORBTION'9X,'REMOVAL'10X,'NU-FISSION'/
     2  2X,'REGION'7X,'(CM)'11X,'(CM)'13X,'(1/CM)'12X,'(1/CM)'12X,
     3'(1/CM)')
      PI = 3.14159
      II = 1
      R = 0.0
      A1(1) = 0.0
      A2(1) = 0.0
      B(1,1,1) = 0.0
      B(1,2,1) = 0.0
      B(2,1,1) = 0.0
      B(2,2,1) = 0.0
      DO 1 N=1,NR
      READ (2,101) D1,D2,SA1,SA2,SR1,SF1,SF2,DR
      WRITE(1,508) N, DR, D1, SA1, SR1, SF1, D2, SA2, SF2
  508 FORMAT(I6,F14.5,4E18.6,/20X,2E18.6,18X,E18.6)
```

```
      NS = INTER(N)
      NE = INTER(N+1)-1
      DO 1 I=NS,NE
      DEL = PI*(2.0*R/DR+1.0)
      A1(I+1) = D1*DEL
      A2(I+1) = D2*DEL
      C1(I) = A1(I+1)
      C2(I) = A2(I+1)
      VOL = PI*DR*(R+DR/4.0)
      B(1,1,I) = D1*DEL+VOL*(SR1+SA1+BZS*D1-SF1)+B(1,1,I)
      B(2,2,I) = D2*DEL+VOL*(SA2+BZS*D2)+B(2,2,I)
      B(1,2,I) = B(1,2,I)-VOL*SF2
      B(2,1,I) = B(2,1,I)-VOL*SR1
      VOL = PI*DR*(R+3.0*DR/4.0)
      B(1,1,I+1) = D1*DEL+VOL*(SR1+SA1+BZS*D1-SF1)
      B(2,2,I+1) = D2*DEL+VOL*(SA2+BZS*D2)
      B(1,2,I+1) = 0.0     -VOL*SF2
      B(2,1,I+1) = 0.0     -VOL*SR1
      RAD(I) = R
      NTEST = NC(II)-I
      IF (NTEST) 3,2,3
    2 IF (I-1) 222,221,222
  221 VOL = PI*DR*DR/4.0
      GO TO 223
  222 VOL = PI*2.0*R*DR
  223 G1(I) = VOL*SP1
      G2(I) = VOL*SP2
      II = II+1
      GO TO 4
    3 G1(I) = 0.0
      G2(I) = 0.0
    4 CONTINUE
      R = R+DR
    1 CONTINUE
      C1(NP) = 1.0
      C2(NP) = 1.0
```

```
C    MAKE GUESS AT EIGENVALUE AND DO WIELANDT TRANSFORMATION
  52 READ (2,102) Z,ITMAX
 102 FORMAT(F10.0,I10)
     WRITE(1,509) Z
 509 FORMAT(///'EIGENVALUE GUESS ='E13.6)
     CALL DATSW(1,II)
     IF (II-1) 67,68,67
  68 E1 = 0.0
     GO TO 69
  67 E1 = 5.E-7
  69 CONTINUE
C    PERFORM WIELANDT TRANSFORMATION AND SET UP CHOLESKI ALGORITHM COEFFICIENTS.
     DO 5 I=1,NP
     W(1,1) = B(1,1,I)+Z*G1(I)-A1(I)*GAM(1,1,I-1)
     W(2,2) = B(2,2,I)+Z*G2(I)-A2(I)*GAM(2,2,I-1)
     W(1,2) = B(1,2,I)-A1(I)*GAM(1,2,I-1)
     W(2,1) = B(2,1,I)-A2(I)*GAM(2,1,I-1)
     IJ = I
     CALL QINV(W,IJ)
     IF (IJ) 74,53,53
  53 GAM(1,1,I) = W(1,1)*C1(I)
     GAM(1,2,I) = W(1,2)*C2(I)
     GAM(2,1,I) = W(2,1)*C1(I)
   5 GAM(2,2,I) = W(2,2)*C2(I)
C    BEGIN ITERATION FOR FLUX VECTOR
     IT = 0
     X = 2*NP
     CALL DATSW(0,II)
     Y = 2*II-3
     Y = X*Y
     DO 61 J=1,NP
     F1(J) = 1.0/Y
  61 F2(J) = 1.0/X
  51 CONTINUE
     IT = IT+1
     DO 6 I=1,NP
```

```
      FS1 = A1(I)*O1(I-1)-F1(I)*G1(I)
      FS2 = A2(I)*O2(I-1)-F2(I)*G2(I)
      O1(I) = GAM(1,1,I)/C1(I)*FS1+GAM(1,2,I)/C2(I)*FS2
    6 O2(I) = GAM(2,1,I)/C1(I)*FS1+GAM(2,2,I)/C2(I)*FS2
      X1(NP) = O1(NP)
      X2(NP) = O2(NP)
      DO 7 I=1,NPM
      II = NP-I
      X1(II) = O1(II)+GAM(1,1,II)*X1(II+1)+GAM(1,2,II)*X2(II+1)
    7 X2(II) = O2(II)+GAM(2,1,II)*X1(II+1)+GAM(2,2,II)*X2(II+1)
C     NORMALIZE THE NEW FLUX VECTOR
      SUM = 0.0
      DO 8 I=1,NP
    3 SUM = SUM+ABS(X1(I))+ABS(X2(I))
      I = 0
   15 I = I+1
      IF (F1(I)) 16,15,17
   16 IF (X1(I)) 18,15,19
   17 IF (X1(I)) 19,15,18
   19 SUM = 0.0-SUM
   18 CONTINUE
      DO 9 I=1,NP
      X1(I) = X1(I)/SUM
    9 X2(I) = X2(I)/SUM
C     TEST FOR CONVERGENCE OF FLUX VECTOR
      DO 10 I=1,NP
      TEST = ABS(X1(I)/F1(I)-1.0)
      IF (TEST-E1) 111,111,11
  111 TEST = ABS(X2(I)/F2(I)-1.0)
      IF (TEST-E1) 10,10,11
   10 CONTINUE
      GO TO 14
   11 IF (IT-ITMAX) 12,21,21
   12 DO 13 I=1,NP
      F1(I) = X1(I)
   13 F2(I) = X2(I)
```

```
      GO TO 51
 14   CONTINUE
C     CONVERGENCE OBTAINED -- COMPUTE EIGENVALUE
      EV = Z+1.0/SUM
      WRITE(1,200) EV, IT
 200  FORMAT(//'EIGENVALUE ='E13.6,5X,'EIGENVECTOR CONVERGED IN' I4,
     1  ' ITERATIONS.')
      GO TO 22
 21   EV = Z+1.0/SUM
      WRITE(1,201) ITMAX, EV
 201  FORMAT(//'*****EIGENVECTOR FAILED TO CONVERGE IN' I4,' ITERATIONS.
     1  EIGENVALUE HAS BECOME'E13.6)
      CALL SSWTCH(3,II)
      IF (II-1) 77,22,77
 77   CONTINUE
      FS1 = F2(1)
      DO 231 I=1,NP
      F1(I) = F1(I)/FS1
 231  F2(I) = F2(I)/FS1
      WRITE(1,209)
 209  FORMAT(///'RADIUS'23X,'FLUX'/1X,'(CM)'15X,'FAST'13X,'THERMAL')
      DO 232 I=1,NP
      CALL SSWTCH(2,II)
      IF (II-1) 232,233,232
 232  WRITE(1,210) RAD(I),F1(I),F2(I)
 210  FORMAT(F6.2,5X,2E18.6)
      GO TO 74
 233  WRITE(1,206)
      GO TO 74
 22   CONTINUE
C     BEGIN ITERATION FOR ADJOINT VECTOR
      IT = 0
      DO 30 J=1,NP
      FA1(J) = 1.0/Y
 30   FA2(J) = 1.0/X
 29   CONTINUE
```

```fortran
      IT = IT+1
      O1(1) = 0.0-G1(1)*FA1(1)
      O2(1) = 0.0-G2(1)*FA2(1)
      DO 31 I=2,NP
      O1(I) = GAM(1,1,I-1)*O1(I-1)+GAM(2,1,I-1)*O2(I-1)-G1(I)*FA1(I)
31    O2(I) = GAM(1,2,I-1)*O1(I-1)+GAM(2,2,I-1)*O2(I-1)-G2(I)*FA2(I)
      X1(NP) = GAM(1,1,NP)*O1(NP)+GAM(2,1,NP)*O2(NP)
      X2(NP) = GAM(1,2,NP)*O1(NP)+GAM(2,2,NP)*O2(NP)
      DO 32 I=1,NPM
      II = NP-I
      FS1 = O1(II)+A1(II+1)*X1(II+1)
      FS2 = O2(II)+A2(II+1)*X2(II+1)
      X1(II) = (GAM(1,1,II)*FS1+GAM(2,1,II)*FS2)/C1(II)
32    X2(II) = (GAM(1,2,II)*FS1+GAM(2,2,II)*FS2)/C2(II)
C     NORMALIZE THE NEW ADJOINT VECTOR
      SUM = 0.0
      DO 33 I=1,NP
33    SUM = SUM+ABS(X1(I))+ABS(X2(I))
40    I = 0
      I = I+1
41    IF (FA1(I)) 41,40,42
42    IF (X1(I)) 43,40,44
44    IF (X1(I)) 44,40,43
44    SUM = 0.0-SUM
43    CONTINUE
      DO 34 I=1,NP
      X1(I) = X1(I)/SUM
34    X2(I) = X2(I)/SUM
C     TEST FOR CONVERGENCE OF ADJOINT VECTOR
      DO 35 I=1,NP
      TEST = ABS(X1(I)/FA1(I)-1.0)
      IF (TEST-E1) 135,135,36
135   TEST = ABS(X2(I)/FA2(I)-1.0)
      IF (TEST-E1) 35,35,36
35    CONTINUE
      GO TO 39
```

```fortran
   36 IF (IT-ITMAX) 37,46,46
   37 DO 38 I=1,NP
      FA1(I) = X1(I)
   38 FA2(I) = X2(I)
      GO TO 29
   39 CONTINUE
C     CONVERGENCE OBTAINED -- COMPUTE EIGENVALUE
      EV = Z+1.0/SUM
      WRITE (1,203) EV, IT
  203 FORMAT(//'EIGENVALUE ='E13.6,5X,'ADJOINT VECTOR CONVERGED IN'I4,
     1 ' ITERATIONS.')
      GO TO 471
   46 EV = Z+1.0/SUM
      WRITE(1,202) ITMAX, EV
  202 FORMAT(//'*****ADJOINT VECTOR FAILED TO CONVERGE IN'I4,
     1 ' ITERATIONS.  EIGENVALUE HAS BECOME' E13.6)
  471 CONTINUE
C     NORMALIZE EIGENVECTOR AND ADJOINT SO THAT THEIR DOT PRODUCT IS UNITY
      FS1 = F2(1)
      DO 451 I=1,NP
      F1(I) = F1(I)/FS1
  451 F2(I) = F2(I)/FS1
      SUM = 0.0
      DO 452 I=1,NP
  452 SUM = SUM+F1(I)*FA1(I)*G1(I)+F2(I)*FA2(I)*G2(I)
      DO 453 I=1,NP
      FA1(I) = FA1(I)/SUM
  453 FA2(I) = FA2(I)/SUM
C     PRINT RESULTS
      WRITE(1,204)
  204 FORMAT(///'RADIUS'23X,'FLUX'33X,'ADJOINT FLUX'/1X,'(CM)'15X,'FAST'
     1' 13X,'THERMAL'17X,'FAST'13X,'THERMAL')
      DO 48 I=1,NP
      CALL SSWTCH(2,II)
      IF (II-1) 48,49,48
   48 WRITE(1,205) RAD(I),F1(I),F2(I),FA1(I),FA2(I)
```

```
  205 FORMAT(F6.2,5X,2E18.6,5X,2E18.6)
      GO TO 50
   49 WRITE(1,206)
  206 FORMAT(/'*****VECTOR PRINTOUT TERMINATED MANUALLY BY SENSE SWITCH
     12.')
   50 CONTINUE
C     DETERMINE IF VECTOR SHOULD BE PUNCHED ON CARDS
      WRITE (1,303)
  303 FORMAT(//'SHOULD VECTORS BE PUNCHED ON CARDS?'/)
      READ (6,105) II
  105 FORMAT(A3)
      IF (II+14878) 74,73,74
   73 WRITE(2,304) EV
  304 FORMAT(/E16.8)
      WRITE(2,305) (F1(I),I=1,NP), (F2(I),I=1,NP)
      WRITE(2,305) (FA1(I),I=1,NP), (FA2(I),I=1,NP)
  305 FORMAT(4E20.8)
   74 CONTINUE
C     DETERMINE IF ANOTHER EIGENVALUE GUESS IS TO BE MADE
      WRITE (1,301)
  301 FORMAT(//        'IF ANOTHER EIGENVALUE GUESS IS TO BE MADE, LOAD
     1CARD IN READER, PUSH READER START'/'SET SENSE SWITCH 0 AND PUSH C
     2OMPUTER START.')
      PAUSE 0001
      CALL SSWTCH(0,II)
      IF (II-1) 70,52,70
   70 CONTINUE
C     DETERMINE IF ANOTHER CORE CONFIGURATION IS TO BE RUN.
      WRITE(1,302)
  302 FORMAT(/'IF ANOTHER CORE CONFIGURATION IS TO BE RUN, LOAD CARDS IN
     1 READER, PUSH READER START'/'SET SENSE SWITCH 1 AND START.')
      PAUSE 0002
      CALL SSWTCH (1,II)
      IF (II-1) 72,71,72
   72 CALL EXIT
      END
```

```
      SUBROUTINE QINV(W,IJ)
C     THIS SUBROUTINE INVERTS A GENERAL 2X2 MATRIX, AFTER CHECKING THE DET.
      DIMENSION W(2,2)
      E1 = 1.E-5
      DENOM = W(1,1)*W(2,2)-W(1,2)*W(2,1)
      SAVE = ABS(DENOM)
      IF (SAVE-E1) 2,2,1
   1  SAVE = W(1,1)
      W(1,1) = W(2,2)/DENOM
      W(1,2) = 0.0-W(1,2)/DENOM
      W(2,1) = 0.0-W(2,1)/DENOM
      W(2,2) = SAVE/DENOM
      GO TO 3
   2  WRITE(1,100) DENOM,IJ
 100  FORMAT('THE MATRIX B' I2,' HAS A DETERMINATE VALUE OF'E9.2,' WHICH
     1 IS TOO SMALL TO GIVE AN ACCURATE INVERSE.'/'MAKE A SLIGHTLY DIFFE
     2RENT EIGENVALUE GUESS AND TRY AGAIN.')
      IJ = 0-2
   3  RETURN
      END
```

APPENDIX III

A DESCRIPTION OF THE "GAUSS" OPTIMIZATION CODE

The GAUSS computer code was written by Kylstra* at the
University of Florida as a general program for finding the
"least squared error" optimum for user specified problems.
The user supplies a subroutine, UFUNT, which calculates the
error functions for his particular problem. He also sup-
plies the auxiliary input and output subroutines, USEGI and
USEGO, which communicate any additional values to be used
in UFUNT.

The main program uses the methods of Gauss and Legendre
for finding the optimum set of variables. Since these
methods are derived in reference 25, only the results will
be quoted here as they apply to the problem of selecting
the optimum poison concentrations.

The normalized power density vector, as calculated in
UFUNT by the poison eigenvector expansion technique, is
subtracted from the flat "ideal power" vector to give an
error vector E. The weighted sum of the squared elements
of this error vector is the scalar error value to be mini-
mized.

---

*C.D. Kylstra, personal communication.

130

$$\text{ERROR} = E = E^T [W] E, \tag{119}$$

where $[W]$ is the matrix of fractional volumes used to weight the power density. The Gaussian optimization technique uses both the scalar error and the error vector. By calling the UFUNT subroutine repeatedly, varying only one poison concentration from a given set each time, an m by n matrix is established giving the variation of each element of E with each poison concentration parameter, $N(j)$. Here, m is the number of power densities computed, and n is the number of variable poison concentrations. This matrix is just a finite difference approximation of the Jacobian matrix, $[J]$, as evaluated at the given set of poison concentrations.[25] Using this matrix, the set of parameters, $\overline{N}$, for which the scalar error will vanish is predicted by the equation

$$\Delta \overline{N} = -\{ [J] [W] [J]^T \}^{-1} [J] [W] E. \tag{120}$$

However, it may be that the predicted optimum, $\overline{N} + \Delta \overline{N}$, lies beyond the region in which the assumptions made in replacing $[J]$ with its finite difference approximation are valid. This may be determined in one of two ways. Either the magnitude of the vector $\Delta \overline{N}$ is greater than some allowed limit, or, when the GAUSS program calls UFUNT to evaluate the scalar error for the parameters $\overline{N} + \Delta \overline{N}$, it finds the error has increased, rather than decreased. In either case, the parameter step size is limited to some smaller value, r, and

the program minimizes the scalar error, E, on the surface of a "hypersphere" in "N space". That is, the problem is constrained such that

$$|\Delta\overline{N}'| = r^2. \tag{121}$$

By writing the Lagrangian for this constrained problem, making a virtual displacement and requiring that the Lagrangian be stationary, the solution may be found as

$$\Delta N' = -\{[J][W][J]^T - \lambda[I]\}^{-1}[J][W]E, \tag{122}$$

where λ is the Lagrange multiplier of the constraint relation (121). The value of λ may be computed for a given step size from the relation

$$r^2 = E^T[J]^T\{[J][W][J]^T - \lambda[I]\}^{-2}[J]E. \tag{123}$$

However, this relation is not used due to the added calculations that it would require. Instead, λ is varied directly. As λ ranges between infinity and zero, r varies between zero and $\Delta\overline{N}$, the predicted optimum step size. Thus, the choice of a large value for λ restricts the search for an optimum to a small hypersphere near the point at which the [J] matrix was evaluated. The Lagrange multiplier, λ, when used in this context, is known as the Levenberg parameter.

Using a large initial value for λ, $\Delta\overline{N}'$ is computed from equation (122). The main GAUSS program then calls UFUNT to evaluate the scalar error for the new parameter set,

$\overline{N}+\Delta\overline{N}'$. If it is indeed less than the error at $\overline{N}$, the program continues to increase the step size, r, by decreasing λ, until either the scalar error begins to increase again, or the limiting value on $\Delta\overline{N}$ is reached. The best value of $\overline{N}+\Delta\overline{N}'$ is then designated the new optimum, about which another search iteration commences with the construction of a new Jacobian. If no point is found for which the scalar error is smaller than the beginning point for the present iteration, the program constructs a new Jacobian about the same point, using a smaller step size to compute the derivatives. This whole process continues until it is determined that the minimum point lies within some specified convergence distance of the last point for which a Jacobian was constructed. The program then outputs the optimum parameter set.

This basic GAUSS program had to be modified in two ways for use with the poison eigenvalue search technique. First, the ability to use a weighting matrix, $[W]$, was incorporated into the program as reflected by the above equations. Also, the program was modified to retain the value of ϵ, the poison eigenvalue, that was calculated for the point, $\overline{N}$, about which the search for $\Delta\overline{N}$ is taking place, so that it may be used as the starting guess at the value of ϵ for all trial poison distributions, $\overline{N}+\Delta\overline{N}$. The value of ϵ must also be saved for the best point found so far. Without this latter modification, the author experienced

some difficulty with the UFUNT subroutine converging to the wrong root of ε. This occurred when the main GAUSS program, returning from an extended Lagrange search to the previously computed optimum point, brought a value of ε_1 from the last calculation in UFUNT that was drastically different from the value of ε_1 associated with the optimum point.

The user supplied subroutines are listed in Appendix IV to complete the description of the optimization program. This modified version of GAUSS works quite well as a demonstration device. However, because this method is limited to the inherently unrealistic squared errors representation, other optimization methods should perhaps be considered if the techniques advocated in this study are to be incorporated into a standardized code to be used as an engineering tool. (The unrealistic aspects of the least squared error optimization are discussed in Chapter VII.)

APPENDIX IV

LISTING OF THE USER SUPPLIED SUBROUTINES FOR
THE USE OF THE "GAUSS" PROGRAM TO FIND THE
OPTIMUM CRITICAL POISON DISTRIBUTION

Partial Dictionary to the Computer Code Variables:

GAUSS Variables	Name or Variable in Text
A(I)	expansion coefficients, a_i
AV(I,J)	elements of adjoint vectors, $\psi_i(j)$
CIP(J)	increments for variation of the poison concentrations, $\Delta N(j)$
ERR	sum of the squared errors
EPINV	$1/\varepsilon$
EV(I,J)	elements of eigenvectors, $\phi_i(j)$
F(M,N)	$[F]$ matrix
FL(L)	flux vector (fast and thermal elements)
FUMB(J)	errors in local power densities
INTER(J)	list of interface indices
ITMAX	maximum number of iterations for coefficients allowed for each poison distribution
MNIP	number of control points
NE	number of elements in flux vector
NIP	number of independently variable poison concentrations
NP	number of flux evaluation points
NR	number of regions
NTP	number of power density evaluations
NV	number of eigenvectors used in the expansion
P(J)	power densities
SF1(M)	list of values for Σ_f^1 in each region
SF2(M)	list of values for Σ_f^2 in each region
SP(J)	diagonal elements of $[\sigma_p]$ matrix
TIP(J)	poison concentrations at control points, $N(j)$
WTF(J)	weighting factors for each control point in error calculations (diagonal elements of $[W]$)
Z(I)	eigenvalues, ε_i, for the eigenvectors

```
      SUBROUTINE USEGI

C     THIS SUBROUTINE IS WRITTEN BY THE USER OF THE GENERAL GAUSS PROGRAM.
C     IT MAY BE USED TO INPUT ADDITIONAL INFORMATION FOR USE IN THE 'UFINT'
C     SUBROUTINE.
C
      COMMON            FUNB(50), TIP(16),          CIP(16), ERR, MNIP,
     $ NIP, NTP,        NID, NOD, MAX
      COMMON F(30,30),Z(30),EV(30,50),AV(30,50),SF1(5),SF2(5),WTF(50),
     $POW(50),FL(100),SP(100),       EPINV,A(30)
      COMMON            P(16), DVEC(16), IFIP(16), TXP(16), COLY(16),
     $ NIP2, NIPT, IMPO1, IMPO2, ICT, IXKC
      COMMON NP,NE,NV,NCP,NR,ITMAX,INTER(5), IND(55)
      COMMON IMPO3, IMPO4
      READ (NID,100) ITMAX,NV,NE,NP,NR
100   FORMAT(16I5)
      READ (NID,100) (INTER(I),I=1,NR)
      DO 1 I=1,NV
      READ (NID,103) Z(I)
103   FORMAT(E16.6)
      READ (NID,101) (EV(I,J),J=1,NE)
101   FORMAT (4E20.6)
      READ (NID,101) (AV(I,J),J=1,NE)
1     CONTINUE
      READ (NID,101) (SF1(I),SF2(I),I=1,NR)
      READ (NID,101) (SP(I),I=1,NE)
      READ (NID,101) EPINV
      IF (IMPO3) 2,3,2
2     WRITE(NOD,104)
104   FORMAT('1CARD IMAGES FOR DATA'///)
      WRITE(NOD,2021) MNIP, NIP, NTP, MAX, IMPO1, IMPO2, IMPO3, IMPO4
      WRITE(NOD,2031) (TIP(I),I=1,MNIP)
      WRITE(NOD,2031) (CIP(I),I=1,MNIP)
      WRITE(NOD,2031) (WTF(I),I=1,NTP)
2021  FORMAT(8I10)
2031  FORMAT(8E15.6)
```

```fortran
      WRITE(NOD,100) ITMAX,NV,NE,NP,NR
      WRITE(NOD,100) (INTER(I),I=1,NR)
      DO 4 I=1,NV
      WRITE(NOD,103) Z(I)
      WRITE(NOD,101) (EV(I,J),J=1,NE)
      WRITE(NOD,101) (AV(I,J),J=1,NE)
    4 CONTINUE
      WRITE(NOD,101) (SF1(I),SF2(I),I=1,NR)
      WRITE(NOD,101) (SP(I),I=1,NE)
      WRITE(NOD,101) EPINV
    3 CONTINUE
      RETURN
      END
```

```fortran
      SUBROUTINE USEGO

C     THIS SUBROUTINE IS WRITTEN BY THE USER OF THE GENERAL GAUSS PROGRAM.
C     IT MAY BE USED TO OUTPUT ADDITIONAL INFORMATION UPON COMPLETION OF
C     THE OPTIMIZATION PROCESS.
C
      DIMENSION XN(60)
      COMMON              FUNB(50), TIP(16),          CIP(16), ERR, MNIP,
     $ NIP, NTP,           NID, NOD, MAX
      COMMON F(30,30),Z(30),EV(30,50),AV(30,50),SF1(5),SF2(5),WTF(50),
     $POW(50),FL(100),SP(100),         EPINV,A(30)
      COMMON         P(16), DVEC(16), IFIP(16), TXP(16), COLY(16),
     $ NIP2, NIPT, IMPO1, IMPO2, ICT, IXKC
      COMMON NP,NE,NV,NCP,NR,ITMAX,INTER(5),  IND(55)
      COMMON IMPO3, IMPO4
      E = 1.0/EPINV
      WRITE (NOD,205) E
  205 FORMAT('0FOR CRITICALITY, MULTIPLY POISON CONCENTRATIONS BY', E13.6
     $)
      WRITE(NOD,210) (POW(I),IND(I),I=1,NTP)
  210 FORMAT('1POWER DISTRIBUTION',5X,'(POINT NUMBER)'/(E19.6,9X,I3))
      II = 0
      DO 2 I=1,NE
      IF (SP(I)) 2,2,4
    4 SUM = 0.0
      DO 1 J=1,NV
    1 SUM = SUM+A(J)*Z(J)*EV(J,I)
      II = II+1
      IND(II) = I
      XN(II) = SUM/FL(I)
    2 CONTINUE
      WRITE(NOD,200)
  200 FORMAT(///,' RECONSTRUCTION OF POISON CONCENTRATIONS GIVES:'//' POI
     $NT NO.',7X,'FAST GROUP',6X,'THERMAL GROUP',10X,'(ACTUAL VALUE)')
      DO 5 I=1,MNIP
    5 TIP(I) = TIP(I)*E
```

```
      DO 3 I=1,MNIP
      II = I+MNIP
    3 WRITE(NOD,201) IND(I), XN(I), XN(II), TIP(I)
  201 FORMAT(I7,3X,2E18.6,10X,E13.6)
      WRITE(NOD,202) (I,FL(I),FL(I+NP),I=1,NP)
  202 FORMAT(////' CONSTRUCTED FLUXES:'//' POINT NO.',7X,'FAST GROUP',6X,
     $       'THERMAL GROUP'/(I7,3X,2E18.6))
      RETURN
      END
```

```
      SUBROUTINE UFUNT

C     THIS SUBROUTINE IS WRITTEN BY THE USER OF THE GENERAL GAUSS PROGRAM.
C     IT IS USED TO CALCULATE THE VALUES OF THE ERROR FUNCTIONS FOR EACH
C     SET OF PARAMETERS SELECTED BY THE MAIN GAUSS OPTIMIZATION PROGRAM.
C
      DIMENSION FF(900), MW(30), LW(30)
      COMMON          FUNB(50), TIP(16),             CIP(16),  ERR,  MNIP,
     $ NIP, NTP,              NID, MOD, MAX
      COMMON  F(30,30),Z(30),EV(30,50),AV(30,50),SF1(5),SF2(5),WTF(50),
     $POW(50),FL(100),SP(100),         EPINV,A(30)
      COMMON          P(16), DVEC(16), IFIP(16), TXP(16), COLY(16),
     $ NIP2, NIPT, IMPO1, IMPO2, ICT, IXKC
      COMMON NP,NE,NV,NCP,NR,ITMAX,INTER(5), IND(55)
      COMMON IMPO3, IMPO4
      EQUIVALENCE           (LW(1),FL(1)),  (MW(1),FL(50))
      DO 3 I=1,NV
      DO 3 J=1,NV
      SUM = 0.0
      II = 0
      DO 2 K=1,NP
      IF (SP(K)) 1,2,1
    1 II = II+1
      KPNP = K+NP
      SUM = SUM+SP(K)*TIP(II)*AV(J,K)*EV(I,K)+SP(KPNP)*TIP(II)
     1 *AV(J,KPNP)*EV(I,KPNP)
    2 CONTINUE
    3 F(J,I) = SUM
      IF (EPINV) 4,4,5
    4 EPINV = F(1,1)/Z(1)
    5 DO 6 I=1,NV
    6 F(I,1) = F(I,1)-EPINV*Z(I)
   17 DO 20 J=1,NV
      DO 20 I=1,NV
      II = (J-1)*NV+I
   20 FF(II) = F(I,J)
```

```
      CALL MINV(FF,NV,DET,LW,MW)
      IF (IMPO4) 26,26,27
   27 WRITE(NOD,299) EPINV
  299 FORMAT(' GUESS AT 1/E =',E13.6)
      WRITE(NOD,506) DET
  506 FORMAT(' DETERMINANT OF F MATRIX =',E13.6  )
   26 IF (DET) 24,22,24
   22 ERR = EPINV*0.00001
      EPINV = EPINV+ERR
      DO 23 I=1,NV
   23 F(I,I) = F(I,I)-ERR*Z(I)
      GO TO 17
   24 DO 25 J=1,NV
      DO 25 I=1,NV
      II = (J-1)*NV+I
   25 F(I,J) = FF(II)
      DO 7 J=1,NV
      A(J) = 1.0
      DO 7 I=1,NV
    7 F(J,I) = F(J,I)*Z(I)
      CALL CONVG(A,EPINV,NV,F,NOD,ITMAX,TIP,MNIP)
      DO 9 I=1,NE
      SUM = 0.0
      DO 8 J=1,NV
    8 SUM = A(J)*EV(J,I)+SUM
    9 FL(I) = SUM
      SUM = 0.0
      II = 0
      IR = 1
      DO 10 I=1,NP
      II = II+1
      IPNP = I+NP
      POW(II) = (SF1(IR)*FL(I)+SF2(IR)*FL(IPNP))
      IND(II) = I
      SUM = SUM+POW(II)*WTF(II)
      NTEST = INTER(IR)-I
```

```fortran
      IF (NTEST) 11,11,10
11    IR = IR+1
      IF (IR-NR) 14,14,10
14    II = II+1
      IPNP = 1+NP
      POW(II) = (SF1(IR)*FL(I)+SF2(IR)*FL(IPNP))
      IND(II) = I
      SUM = SUM+POW(II)*WTF(II)
10    CONTINUE
      SUM = ABS(SUM)
      DO 13 I=1,II
13    POW(I) = POW(I)/SUM
      ERR = 0.0
      DO 12 I=1,NTP
      FUNB(I) = POW(I)-1.0
12    ERR = ERR+FUNB(I)*FUNB(I)*WTF(I)
      IF (IMPO4) 15,16,15
15    WRITE(NOD,500) (TIP(I),I=1,MNIP)
      WRITE(NOD,500) (FL(I),I=1,NE)
      WRITE(NOD,500) (A(I), I=1,NV)
      WRITE(NOD,500) (POW(I),I=1,NTP)
      WRITE(NOD,500) (FUNB(I),I=1,NTP)
500   FORMAT(5E20.6)
16    CONTINUE
      WRITE(NOD,500) ERR, EPINV
      RETURN
      END
```

```fortran
      SUBROUTINE CONVG(VEC,VAL,NN,D,NOD,MAX,TIP,MNIP)
      DIMENSION D(30,30), VEC(30), V2(30), TIP(16)
      E1 = 5.E-6
      L = 1
      ON = NN
      E2 = 0.0001/ON
    1 DO 3 I=1,NN
      SUM = 0.0
      DO 2 J=1,NN
    2 SUM = SUM+D(I,J)*VEC(J)
    3 V2(I) = SUM
      VN = 0.0
      TEST = 0.0
      DMAX = 0.0
      DO 5 I=1,NN
      ABV = ABS(VEC(I))
      IF (ABV-E2) 5,5,20
   20 VE = V2(I)/VEC(I)
      VN = VE*ABV+VN
      TEST = ABS(VE/ON-1.0)
      IF (TEST-DMAX) 5,5,4
    4 DMAX = TEST
    5 CONTINUE
      IF (ABS(VN)-E1) 22,22,23
   22 VN = ON
   23 DO 21 I=1,NN
   21 VEC(I) = V2(I)/VN
      ON = VN
      IF (DMAX-E1) 10,10,7
    7 IF (MAX-L) 9,9,8
    8 L = L+1
      GO TO 1
    9 WRITE(NOD,101) DMAX, VAL,    (TIP(I),I=1,MNIP)
  101 FORMAT('   CONVERGENCE NOT OBTAINED FOR EPSILON, DMAX =',E13.6,',',
     $EPINV =',E13.6,',  THE TIP TABLE IS:',/(5E15.6))
   10 VAL = VAL+1.0/VN
```

```
      VN = VEC(1)
      DO 12 I=1,NN
12 VEC(I) = VEC(I)/VN
      RETURN
      END
```

```
      SUBROUTINE MINV(A,N,D,L,M)                              MINV    1
      DIMENSION A(1),L(1),M(1)                                MINV    2
C        SEARCH FOR LARGEST ELEMENT                           MINV    3
      D=1.0                                                   MINV    4
      NK=-N                                                   MINV    5
      DO 80 K=1,N                                             MINV    6
      NK=NK+N                                                 MINV    7
      L(K)=K                                                  MINV    8
      M(K)=K                                                  MINV    9
      KK=NK+K                                                 MINV   10
      BIGA=A(KK)                                              MINV   11
      DO 20 J=K,N                                             MINV   12
      IZ=N*(J-1)                                              MINV   13
      DO 20 I=K,N                                             MINV   14
      IJ=IZ+I                                                 MINV   15
   15 IF( ABS(BIGA)- ABS(A(IJ))) 15,20,20                     MINV   16
   15 BIGA=A(IJ)                                              MINV   17
      L(K)=I                                                  MINV   18
      M(K)=J                                                  MINV   19
   20 CONTINUE                                                MINV   20
C        INTERCHANGE ROWS                                     MINV   21
      J=L(K)                                                  MINV   22
      IF(J-K) 35,35,25                                        MINV   23
   25 KI=K-N                                                  MINV   24
      DO 30 I=1,N                                             MINV   25
      KI=KI+N                                                 MINV   26
      HOLD=-A(KI)                                             MINV   27
      JI=KI-K+J                                               MINV   28
      A(KI)=A(JI)                                             MINV   29
   30 A(JI) =HOLD                                             MINV   30
C        INTERCHANGE COLUMNS                                  MINV   31
   35 I=M(K)                                                  MINV   32
      IF(I-K) 45,45,38                                        MINV   33
   38 JP=N*(I-1)                                              MINV   34
      DO 40 J=1,N                                             MINV   35
      JK=NK+J                                                 MINV   36
```

```
      JI=JP+J                                                       MINV  37
      HOLD=-A(JK)                                                   MINV  38
      A(JK)=A(JI)                                                   MINV  39
   40 A(JI) =HOLD                                                   MINV  40
C         DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT IS   MINV  41
C         CONTAINED IN BIGA)                                        MINV 103
   45 IF(ABS(BIGA)-1.E-20)45,46,48                                  MINV  44
   46 D=0.0                                                         MINV  45
      RETURN                                                        MINV  46
   48 DO 55 I=1,N                                                   MINV  47
      IF(I-K) 50,55,50                                              MINV  48
   50 IK=NK+1                                                       MINV  49
      A(IK)=A(IK)/(-BIGA)                                           MINV  50
   55 CONTINUE                                                      MINV  51
C         REDUCE MATRIX                                             MINV  52
      DO 65 I=1,N                                                   MINV  53
      IK=NK+1                                                       MINV 001
      HOLD=A(IK)                                                    MINV  54
      IJ=I-N                                                        MINV  55
      DO 65 J=1,N                                                   MINV  56
      IJ=IJ+N                                                       MINV  57
      IF(I-K) 60,65,60                                              MINV  58
   60 IF(J-K) 62,65,62                                              MINV  59
   62 KJ=IJ-I+K                                                     MINV 002
      A(IJ)=HOLD*A(KJ)+A(IJ)                                        MINV  61
   65 CONTINUE                                                      MINV  62
C         DIVIDE ROW BY PIVOT                                       MINV  63
      KJ=K-N                                                        MINV  64
      DO 75 J=1,N                                                   MINV  65
      KJ=KJ+N                                                       MINV  66
      IF(J-K) 70,75,70                                              MINV  67
   70 A(KJ)=A(KJ)/BIGA                                              MINV  68
   75 CONTINUE                                                      MINV  69
C         PRODUCT OF PIVOTS                                         MINV  70
      D=D*BIGA                                                      MINV  71
C         REPLACE PIVOT BY RECIPROCAL
```

```
      A(KK)=1.0/BIGA                                          MINV  72
   80 CONTINUE                                                MINV  73
C        FINAL ROW AND COLUMN INTERCHANGE                     MINV  74
      K=N                                                     MINV  75
  100 K=(K-1)                                                 MINV  76
      IF(K) 150,150,105                                       MINV  77
  105 I=L(K)                                                  MINV  78
      IF(I-K) 120,120,108                                     MINV  79
  108 JQ=N*(K-1)                                              MINV  80
      JR=N*(I-1)                                              MINV  81
      DO 110 J=1,N                                            MINV  82
      JK=JQ+J                                                 MINV  83
      HOLD=A(JK)                                              MINV  84
      JI=JR+J                                                 MINV  85
      A(JK)=-A(JI)                                            MINV  86
  110 A(JI) =HOLD                                             MINV  87
  120 J=M(K)                                                  MINV  88
      IF(J-K) 100,100,125                                     MINV  89
  125 KI=K-N                                                  MINV  90
      DO 130 I=1,N                                            MINV  91
      KI=KI+N                                                 MINV  92
      HOLD=A(KI)                                              MINV  93
      JI=KI-K+J                                               MINV  94
      A(KI)=-A(JI)                                            MINV  95
  130 A(JI) =HOLD                                             MINV  96
      GO TO 100                                               MINV  97
  150 RETURN                                                  MINV  98
      END                                                     MINV  99
```

LIST OF REFERENCES

1. Harker, W.H., "Designing Reactor Cores to Achieve Pre-determined Power Shapes," Transactions of the American Nuclear Society: 1, 142-144 (1958).

2. Haling, R.K., "Operating Strategy for Maintaining an Optimum Power Distribution Throughout Life," Proceedings of the Topical Meeting on Nuclear Performance of Power Reactor Cores, 205, San Francisco, American Nuclear Society (September, 1963).

3. Crowther, R.L., "The Power-Control Method for Solution of Diffusion Equations Describing Large Non-Linear Inhomogeneous Reactors," Transactions of the American Nuclear Society: 6, 276-277 (1963).

4. Motoda, H. and Kawai, T., "A Theory of Control-Rod Programming Optimization in Two-Region Reactors," Nuclear Science and Engineering: 39, 114-118 (1970).

5. Terney, W.B. and French, H., "Control Rod Programming Optimization Using Dynamic Programming," Nuclear Science and Engineering: 39, 109-114 (1970).

6. Wade, D.C. and Terney, W.B., "Optimal Control of Nuclear Reactor Depletion," Nuclear Science and Engineering: 45, 199-217 (1971).

7. Jackson, W.R., "An Eigenfunction Approach to the Problem of Reactor Poison Distribution," Master's Thesis, University of Florida, Gainesville (1969).

8. Foderaro, A. and Garabedian, H.L., "A New Method for the Solution of Group Diffusion Equations," Nuclear Science and Engineering: 8, 44-52 (1960).

9. Garabedian, H.L. and Thomas, D.H., "An Analytic Approach to Two Dimensional Reactor Theory," Nuclear Science and Engineering: 14, 266-271 (1962).

10. Hara, F. and Shibata, H., "A Study of Core Optimization from the View Point of Neutron Flux Distribution," Journal of Nuclear Science and Technology: 5, 323-332 (July, 1968).

11. Kaplan, S., "The Property of Finality and the Analysis of Problems in Reactor Space-Time Kinetics by Various Modal Expansions," Nuclear Science and Engineering: 9, 357-361 (1961).

12. Kaplan, S. and Yasinsky, J.B., "Natural Modes of the Xenon Problem with Flow Feedback--An Example," Nuclear Science and Engineering: 25, 430-438 (1966).

13. Henry, A.F., "The Application of Inhour Modes to the Description of Nonseparable Reactor Transients," Nuclear Science and Engineering: 20, 338-351 (1964).

14. Ewen, R.L., "MULE--A Fortran Program for the Calculation of Three Types of Overtone Modes," WAPD-TM-369 (October, 1963).

15. Sommerfeld, A., Partial Differential Equations in Physics, Academic Press, New York (1949).

16. Clark, M., Jr. and Hansen, K.F., Numerical Methods of Reactor Analysis, Academic Press, New York (1964).

17. Shilov, G.E., An Introduction to the Theory of Linear Spaces, Prentice-Hall, Inc., Englewood Cliffs, N.J. (1961).

18. White, P.A., "The Computation of Eigenvalues and Eigenvectors of a Matrix," J. Soc. Ind. Appl. Math: 6, 393-437 (1958).

19. Wielandt, H., "Bestimmung höherer Eigenwerte durch gebrochene Iteration," Bericht der aerodynam ischen Versuchsanstalt Göttingen, Report 44/J/37 (1944).

20. Hansen, K.F. and Clark, M., Jr., "Adjoint Functions and Orthogonality Relations," Nuclear Science and Engineering: 15, 139-141 (1963).

21. Wachspress, E.L., "A Numerical Technique for Solving Group Diffusion Equations," Nuclear Science and Engineering: 8, 164-170 (1960).

22. Varga, R.S., "Numerical Methods for Solving Multi-Dimensional Multigroup Diffusion Equations in Nuclear Reactor Theory," Proceedings of Symposia in Applied Mathematics, Vol. XI, G. Birkhoff, E. P. Wigner, Eds., 164-189, American Mathematical Society, Providence (1961).

23. Anderson, E.C. and Putnam, G.E.,"CORA--A Few Group
    Diffusion Theory Code for One-Dimensional Reactor
    Analysis," IN-1416 (August, 1970).

24. Neal, D.E. et al., "The ARC System One-Dimensional
    Diffusion Theory Capability, DARC1D," Argonne
    National Laboratory, Argonne, Illinois, ANL-7715
    (May, 1971).

25. Wilde, D.J., and Beightler, C.S., Foundations of
    Optimization, Prentice-Hall, Inc., Englewood
    Cliffs, N.J. (1967).

BIOGRAPHICAL SKETCH

Steven Michael Long was born in Bethesda, Maryland, on May 22, 1946. He was graduated from Walter Johnson High School in June, 1964. In June, 1968, he graduated with high honors from Princeton University, receiving his bachelor's degree in aeronautical engineering. Upon graduation, he was inducted into the Phi Beta Kappa honorary scholastic fraternity. The award of an AEC Special Fellowship provided the financial support necessary for him to enter the University of Florida Graduate School in September, 1968. In December, 1969, he received a Master of Science degree in the Department of Nuclear Engineering Sciences. In continuing his studies at the University of Florida, pursuant to the degree of Doctor of Philosophy, he has held a College of Engineering Fellowship and an assistantship with the nuclear engineering department. On June 17, 1972, he married the former Diana Dombey of St. Petersburg, Florida. While at the University of Florida, he has been a student member of the American Nuclear Society.
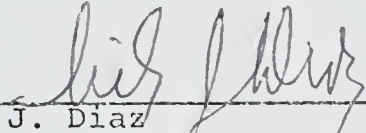
I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

G.R. Dalton, Chairman
Professor of Nuclear Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

M.J. Ohanian
Professor of Nuclear Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

N.J. Diaz
Assistant Professor of Nuclear
                    Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

U.H. Kurzweg
Associate Professor of Engineering
            Sciences and Mechanics

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

D.T. Williams
Professor of Aerospace Engineering

This dissertation was submitted to the Dean of the College of Engineering and to the Graduate Council, and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

December, 1972

Dean, College of Engineering

Dean, Graduate School